



**Ricardo Ângelo dos
Santos Filipe**

Teleoperação de um veículo da Eco-Maratona





**Ricardo Ângelo dos
Santos Filipe
(a28521@ua.pt)**

Teleoperação de um veículo da Eco-Maratona

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Dr. Bernardo Cunha, Professor do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

Este trabalho é dedicado a todos aqueles que fizeram e fazem parte da minha vida e que, de uma forma ou de outra, contribuíram para me transformar naquilo que hoje sou.

o júri

presidente

Doutor João Nuno Pimentel da Silva Matos
Professor Associado da Universidade de Aveiro

Vogais

Doutor Manuel Bernardo Salvador Cunha (Orientador)
Professor Auxiliar da Universidade de Aveiro

Doutor Artur José Carneiro Pereira (Co-Orientador)
Professor Auxiliar da Universidade de Aveiro

Doutor Carlos Baptista Carneira
Professor Auxiliar do Departamento de Engenharia Mecânica do Instituto Superior Técnico da
Universidade Técnica de Lisboa

agradecimentos

Agradeço a todas as pessoas envolvidas, até à data, no projecto ICARO por tudo o que desenvolveram e ajudaram a construir.

De um ponto de vista mais particular, os meus sinceros agradecimentos aos Professores Bernardo Cunha e Manuel Violas, aos Engenheiros Nuno Ferreira, Daniel Castanheira, David Ribeiro e André Neto pelo incansável apoio, disponibilidade e ajuda nas diversas áreas abrangidas por esta dissertação.

palavras-chave

Sistemas electrónicos, monitorização, controlo, teleoperação e telemetria.

resumo

O ICARO é um projecto pluridisciplinar da Universidade de Aveiro, que ao enquadrar trabalhos de áreas tão diversas como mecânica, sistemas de informação e informática, permitem a participação anual de uma equipa da universidade na prova Eco-Maratona Shell que decorre, este ano, em Nogaro, França.

O trabalho desenvolvido em anos subsequentes já permitiu acumular um *know-how* considerável, nomeadamente acerca do modelo arquitectural e funcional dos elementos base, inerentes à monitorização e controlo do veículo.

No que diz respeito a esta tese em particular, desenvolveram-se os módulos Visão Remota, Monitorização e Suporte ao condutor, bem como algumas alterações nos módulos Volante e Controlo/GatewayPalm, além da criação dos *setups* finais para colocação no veículo, quando em condução remota, para controlo da direcção e travão.

Relativamente ao módulo Visão Remota, este é constituído por uma mini câmara a cores, bem como todo o sistema de amplificação e transmissão do sinal, realizado através de uma conexão RF na banda de *UHF* no emissor (instalado no veículo) e um receptor de TV (instalado nas boxes da pista).

De forma a obter uma eficaz monitorização e suporte ao condutor foi criado um programa de software e instalado num *palmtop/PDA*. A electrónica de suporte que lhe está associada irá permitir a transferência da informação do veículo e do volante para o *palmtop/PDA*.

keywords

Electronic systems, monitoring, control, teleoperation, telemetry.

abstract

ICARO is a multidisciplinary project from University of Aveiro, incorporating works from areas such as mechanics, electronics, information systems and computing. The main objective is the annual participation from a team of our university in the Éco-Marathon Shell competition which is held in France.

The work developed in the previous years already permitted to accumulate a considerable *know-how* about the architectural and functional model of the fundamental elements constituents of the monitoring and control systems of the vehicle.

In this thesis in particular we present the details related to the development of a set of modules for remote vision, monitorization and support to the driver, as also some changes in the steering wheel module and Control/GatewayPalm module and the creation of the final setups to be included in the vehicle, while in remote driving mode, for control of direction and brake.

In what concerns to the remote vision, consists of the mini cam , as well of all the amplification ad transmission system of the signal, with a RF link in the band of UHF (in the vehicle) and the TV receptor (in the boxes).

The module of monitorization and support to the driver via *Palmtop/PDA* has been created by developing the software installed in *palmtop/PDA* and the electronics associated with the transmission of data from the vehicle and steering wheel to *palmtop/PDA*.

Conteúdo

Capítulo 1	7
Introdução ao Sistema	7
Arquitectura geral do sistema (ICARO) / estado de arte [1]	7
Arquitectura proposta no ano anterior para o desenvolvimento do sistema de condução remota e de condução normal.....	9
Capítulo 2	13
Desenvolvimento de um sistema de visão remota que permita acompanhar a prova a partir do exterior do veículo	13
Introdução.....	13
Pré-requisitos	13
Sistema do ano transacto.....	15
Considerações teóricas com novo modelo de radiação	15
Sistema Actual.....	17
Sistema de emissão	17
Resultados Finais.....	28
Sistema de Recepção.....	29
Conclusões	30
Capítulo 3	31
Alterações efectuadas no volante.....	31
Introdução.....	31
Alterações efectuadas - causas:	32
Cálculo da aceleração:.....	32
Criação do módulo Giroscópio:	32
Alteração na taxa de amostragem	36
Alteração no monitor ICARO	37
Alteração na tabela das prioridades [1]:	37
Alteração do comando - activação de controlo remoto	38
Conclusões	38
Capítulo 4	39
Projecto e desenvolvimento de um sistema de actuação controlado por um microcontrolador capaz de conduzir e travar o veículo	39
Introdução ao sistema.....	39

Travão.....	40
Alterações no código.....	44
Conclusões	45
Capítulo 5	47
Software de suporte em palmtop/PDA para arquivo local de informação e disponibilização de dados críticos ao piloto	47
Introdução teórica.....	47
Tecnologias de Suporte à comunicação	47
Sistema anterior – PalmOS.....	48
Constituição das tramas existentes nas redes sem fios	49
Arquitectura do sistema.....	50
Piconet – palmtop/PDA como master, volante e módulo de controlo slaves	51
Piconet – Módulo de controlo master, palmtop/PDA e volante slave	52
Scatternet – Volante Master – PDA slave e PDA Master-MC slave	53
Mudança da topologia da rede	54
Programa Final do palmtop/PDA	55
Aspectos gerais.....	55
Ligação Bluetooth e obtenção dos valores para posterior processamento	56
Processamento das tramas	57
Processamento dos dados	58
Aspectos do programa principal	58
Conclusões.....	66
Capítulo 6	67
Módulo GatewayPalm.....	67
Aspectos Gerais.....	67
Introdução teórica.....	67
Rede CAN.....	67
GatewayPalm - Hardware	69
GatewayPalm - Software.....	69
Testes efectuados	70
Conclusões	71
Capítulo 7	73
Conclusões gerais	73

Referências.....	74
Anexo A	75
Anexo B	77
Anexo C	81

Índice de Figuras

figura 1 - Condução normal (volante dentro do carro) [1].	9
figura 2 - Condução remota (volante fora do veículo) [2].	9
figura 3 - Programa monitor ICARO.	11
figura 5 - Pista de Nogaro.	14
figura 4-Relação entre a SNR e a qualidade da imagem [1].	14
figura 6 - Diagrama de blocos 1- sistema emissor.	15
figura 7 - Diagrama de blocos 2 – sistema emissor.	15
figura 8 - Reflexões provocadas pelo chão[3].	15
figura 9 - Modelo meio livre vs modelo efeito altura das antenas.	16
figura 10 - Foto do modulador 2 com pico de -8 dBm e centrado em 570 Mhz.	18
figura 11 - Antena monópolo – do ano transacto.	19
figura 12 - Distância M2 + amplificador (IEETA na imagem).	20
figura 13 - Modulador 2 + amp. GaAs.	21
figura 14 - Esquema do filtro passa-banda.	22
figura 15 - Simulação do filtro UHF.	22
figura 16 - Filtro UHF, resultado em laboratório.	23
figura 17 - Filtro UHF.	23
figura 18 - Antena monópolo simulada.	24
figura 19 - Gráfico do monópolo simulado.	25
figura 20 -Gráfico do monopolo real.	25
figura 21 – Nova antena.	26
figura 23 -Amplificador de potência.	27
figura 22 - Circuito do amplificador de potência.	27
figura 24 - Aspecto real - amplificador de potência.	28
figura 25 - Estado final do sistema de emissão.	28
figura 26 - Sistema de recepção.	29
figura 27 - Aspecto geral do volante [1].	31
figura 28 - Sinal do giroscópio a velocidade constante.	34
figura 29 - Movimento realizado.	35
figura 30 - Sinal proveniente do giroscópio, movimento conhecido à partida.	35
figura 31 - Gráfico do Filtro FIR.	36
figura 32 - Prioridades das ligações.	37
figura 33 - PCB do módulo de controlo. [1].	39
figura 34 - (a) Vista lateral (b) Vista Isométrica.	40
figura 35 – Aspecto do travão na realidade.	41
figura 36 - Modelo do travão utilizado.	42
figura 37 - Modelo da direcção utilizado.	42
figura 38 - Valores de variação de PWM na direcção.	43
figura 39 - Esquema da criação do PWM.	43

figura 40 - Imagem do programa monitor, em PalmOS.....	48
figura 41 - Ligação sem fios bluetooth do sistema – <i>scatternet</i>	50
figura 42 - Programa de testes.....	51
figura 43 - Criação da rede.....	52
figura 44 - Arquitectura global ICARO com a nova topologia.	54
figura 45 - Nova topologia da rede – mais detalhada.....	55
figura 46 - Monitor ICARO.....	56
figura 47 - Aspecto da recepção de dados.....	56
figura 48 - Máquina de estados - <i>process_data</i>	57
figura 49 - Máquina de estados - <i>extract_data</i>	58
figura 50 - Máquina de estados - programa principal.....	60
figura 51 - Menu Principal.....	62
figura 52 - Menu Configuração Bluetooth.	62
figura 53 - Menu Monitor/ <i>replay</i>	62
figura 54 - Menu Altitudes (acedido somente do menu monitor/ <i>replay</i>).	63
figura 55 - <i>Palmtop/PDA</i> em cima - espelho para visualizar a informação[1].	64
figura 56 - Trama CAN.....	68
figura 57 - Prioridade sobre o barramento.	68

Capítulo 1

Introdução ao Sistema

ICARO é um projecto pluridisciplinar da Universidade de Aveiro que enquadra trabalhos de diferentes áreas, nomeadamente mecânica, electrónica, sistemas de informação e informática. Tem por objectivo principal a participação anual de uma equipa da universidade na prova Eco-Maratona Shell.

O trabalho desenvolvido ao longo dos anos, permitiu acumular um conjunto de saberes variados acerca do modelo arquitectural e funcional dos elementos fundamentais constituintes do sistema de monitorização e controlo do veículo. O volante, criado no ano anterior e (praticamente) funcional permite, ao piloto, ligar e desligar o motor, buzinar e actuar sobre o acelerador.

É com o intuito de contribuir para este projecto maior que se enquadra a análise e discussão do trabalho levado a cabo nesta dissertação.

Arquitectura geral do sistema (ICARO) / estado de arte [1]

O ICARO assenta numa arquitectura distribuída com base numa rede CAN¹. No estado de arte em que o referido projecto foi encontrado este era constituído por oito diferentes módulos:

Volante: módulo de interface entre o condutor e o veículo. Este sistema permite a condução remota/não remota do veículo. Possui vários botões de controlo capazes de produzir comandos, como por exemplo, ligar o motor e buzinar. A existência de um potenciómetro permite ao condutor indicar a velocidade desejada. De forma a saber qual o ângulo de rotação que o condutor deseja imprimir na direcção existe um acelerómetro e um manípulo de travagem que são utilizados apenas em condução remota.

Módulo de transmissão de visão remota: permite ao condutor, quando em condução remota, ter uma visão aproximada daquela que teria se se encontrasse no interior do veículo.

Módulo de controlo da direcção e travão/GatewayPalm: quando em modo de condução remota, este sistema controla a direcção e o travão, servindo também de *gateway* de comunicação entre o volante e os restantes módulos do veículo. A tecnologia de comunicação utilizada pelo volante é o bluetooth. Em oposição a esta forma de comunicação estão os restantes módulos do veículo, uma vez que utilizam CAN.

¹ Controllor Area Network

Além disso, este módulo é responsável pelo envio, para *logging* no *palmtop/PDA*¹, de toda a informação existente nos módulos CAN, quer em condução remota quer em condução normal.

Gateway Boxes/Telemetria: permite que o veículo envie informação para registo nas boxes e o envio/recepção de comandos para a condução remota deste.

GPS: Global Positioning System.

Roda: módulo no qual se realiza a leitura de um sensor que gera impulsos², os quais permitem a determinação da velocidade instantânea do veículo e respectivas grandezas derivadas tais como a distancia percorrida.

Motor monitor: é o responsável pela aquisição de valores dos sensores ligados ao motor.

Motor control: módulo responsável pelo controlo do motor.

¹ Programa desenvolvido este ano.

² Número de impulsos por unidade de tempo dependente da velocidade de rotação angular da roda.

Arquitectura proposta no ano anterior para o desenvolvimento do sistema de condução remota e de condução normal

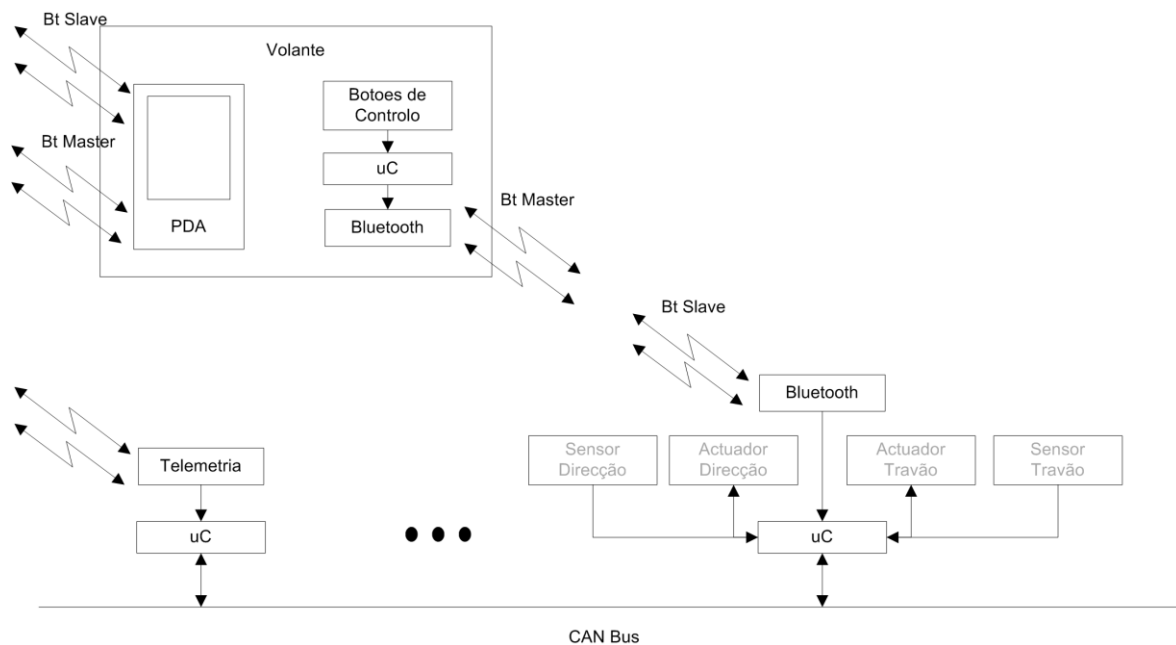


figura 1 - Condução normal (volante dentro do carro) [1].

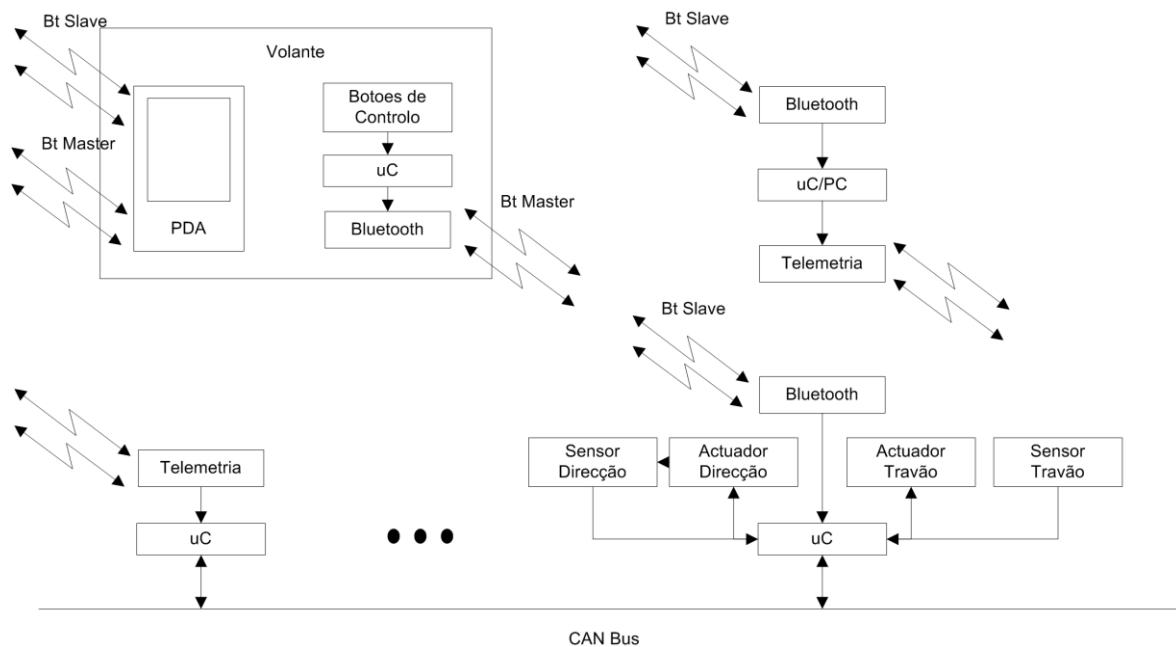


figura 2 - Condução remota (volante fora do veículo) [2].

Tal como se pode verificar pelas figuras 1 e 2 acima apresentadas, a solução proposta no ano anterior inclui dois modos de funcionamento do veículo da EcoMaratona:

Modo de condução normal: que estabelece uma ligação do volante para o módulo de controlo, servindo o *Palmtop/PDA* como *gateway* entre os 2 dispositivos. É condição essencial que o *palmtop/PDA* seja capaz de estabelecer e manter 2 ligações em simultâneo.

A informação proveniente da rede CAN é retransmitida ficando, desta forma, assegurado o registo de dados no *palmtop/PDA* e no PC das boxes, através da Telemetria.

Modo de condução remota

Existem múltiplas formas de se criar a rede:

- Volante <-> PDA <-> módulo de controlo/gateway Palm
- Volante <-> módulo de controlo/gateway Palm
- Volante <-> PDA <-> PC <-> Telemetria
- Volante <-> PC <-> Telemetria

No primeiro caso o *palmtop/PDA* poderá servir de *gateway* entre o volante e o módulo de controlo. No segundo caso, o volante liga-se directamente ao módulo de controlo. Em qualquer uma das opções apresentadas, o accionar da condução remota poderá ser despoletado perto do veículo, desde que os dispositivos bluetooth estejam ao alcance. De forma a activar a condução remota do veículo, é necessário executar um gesto pré-determinado. Esse gesto consiste na colocação do volante a 90 graus, relativamente ao tronco do piloto, colocando o volante novamente na posição inicial.

Com este gesto, os responsáveis pelo accionamento do travão e da direcção entram em funcionamento, permitindo assim o controlo correcto do veículo.

No caso do volante se encontrar nas boxes e no que diz respeito ao terceiro caso, é possível enviar informação para o *palmtop/PDA*, uma vez que este será o responsável pelo envio para o PC. Este, por sua vez, reenviará, por telemetria, para o veículo.

No último caso apresentado poder-se-á enviar a informação (comandos) para o PC, sendo depois enviados, por telemetria, para o veículo.

Em qualquer um destes 2 últimos casos, não é necessário fazer nenhum gesto, pois os comandos provêm do módulo de telemetria que despoleta automaticamente a condução remota.

O programa existente no PC das boxes e que permite enviar os comandos para o veículo é o programa monitor ICARO. Este possui a capacidade de receber dados da rede CAN do veículo e de os visualizar em tempo-real ou ver posteriormente para eliminar possíveis erros do piloto, de condução, entre outros. Além disso o programa permite

também enviar para o veículo os comandos de controlo, responsáveis pela condução remota.

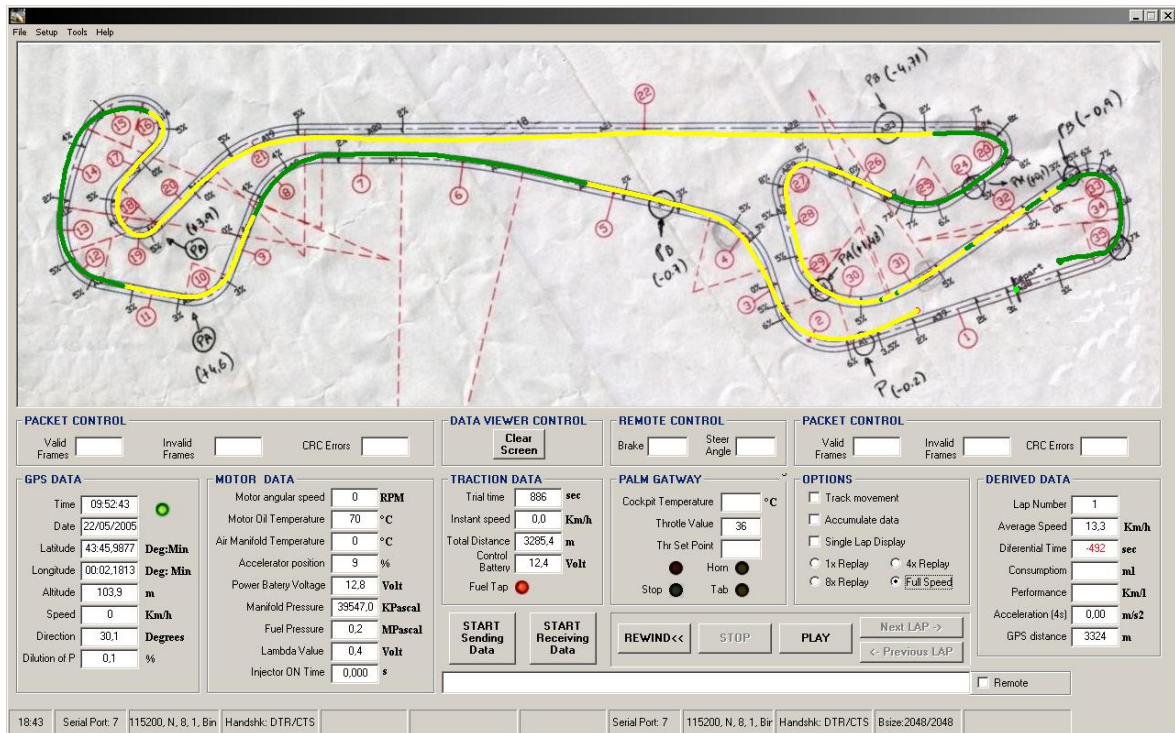


figura 3 - Programa monitor ICARO.

É de referir também que o volante, para além dos botões de controlo, contém também um sensor que permite determinar o ângulo que ele próprio descreve, bem como a quantidade de força que será imprimida no travão. Esta informação será apenas usada quando o veículo se encontrar em modo de condução remota.

Os restantes módulos CAN comportam-se de maneira idêntica em qualquer dos casos, sendo a excepção o módulo de controlo. Este módulo poderá enviar a informação para *logging* no *palmtop/PDA* em qualquer dos casos, desde que este esteja presente.

É neste contexto que se enquadra esta dissertação a qual tem como objectivos principais:

- o desenvolvimento de um sistema de visão remoto que permita acompanhar a prova a partir do exterior do veículo;
- projecto e desenvolvimento de um sistema de actuação controlado por um microcontrolador capaz de conduzir e travar o veículo;
- o desenvolvimento de algoritmos, baseados em processamento digital de imagem, com vista à integração de mecanismos automáticos de suporte à segurança, nomeadamente em situações de falha de comunicação entre a box e o veículo;
- a optimização do software de suporte em *palmtop/PDA* para arquivo local de informação e disponibilização de dados críticos ao piloto;

Capítulo 2

Desenvolvimento de um sistema de visão remota que permita acompanhar a prova a partir do exterior do veículo

Para a teleoperação remota de um veículo, é necessário utilizar um sistema de visão remota que permita a transmissão, em tempo-real, de imagens para um local de controlo do veículo.

Neste sentido, este capítulo pretende dar ênfase ao trabalho desenvolvido nesta área, referindo os componentes do sistema final e evoluções referentes ao ano transacto, bem como os testes de campo e laboratório vs simulações, baseadas em modelos de propagação de radiação.

Introdução

Pré-requisitos

Este sistema foi desenvolvido para a EcoMaratona que, em 2008, decorre em Nogaro, França. Devido ao elevado número de equipas participantes na prova que utilizarão a banda *industrial, scientific and medical* (ISM) que se localiza no espectro de 2.400–2.500 Ghz, adoptou-se a banda *Ultra High Frequency* (UHF) , localizando-se esta entre os 470 e os 806 Mhz.

Cada canal [2] possui uma banda de 6 Mhz, existindo desde o canal 14 ao 69, reservado para televisão, sendo excepção o canal 37 – que é reservado para radioastronomia (608 MHz a 614 MHz). Além das razões acima mencionadas acima, existe também a possibilidade de utilizar produtos já existentes no mercado para esta gama de frequências, o que é sempre um aspecto positivo a ter em conta.

Devido aos aspectos topológicos do circuito (1.2 km de comprimento), o sistema terá de conseguir transmitir, no mínimo, a essa distância¹ com uma qualidade de imagem aceitável.

Nas simulações efectuadas em Matlab [1] adoptou-se como SNR² pretendido o valor de 40 dB para uma imagem com qualidade recebida nas boxes. De forma a ilustrar essas simulações, apresentar-se-ão, de seguida, as imagens relativas a diferentes tipos de SNR e o seu impacto na qualidade da imagem recebida.

¹ A essa distância de raio, pois o sistema não poderá ser directivo, já que o carro não se encontra fixo

² SNR - Relação Sinal-Ruído.



(a) Imagem Original.



(b) Imagem com 40 dB de SNR.



(c) Imagem com 30 dB de SNR.



(d) Imagem com 28 dB de SNR.

figura 4-Relação entre a SNR e a qualidade da imagem [1].

Todavia, o sistema receptor – as boxes, localiza-se num dos extremos da pista, sendo esse o ponto exactamente oposto ao local mais baixo da pista, apresentando um desnível de cerca de 4 metros, em relação às boxes.

Devido a esta barreira natural assumiu-se, então, que seria necessária uma potência que conseguisse transmitir as imagens a uma distância de 3 kms¹.

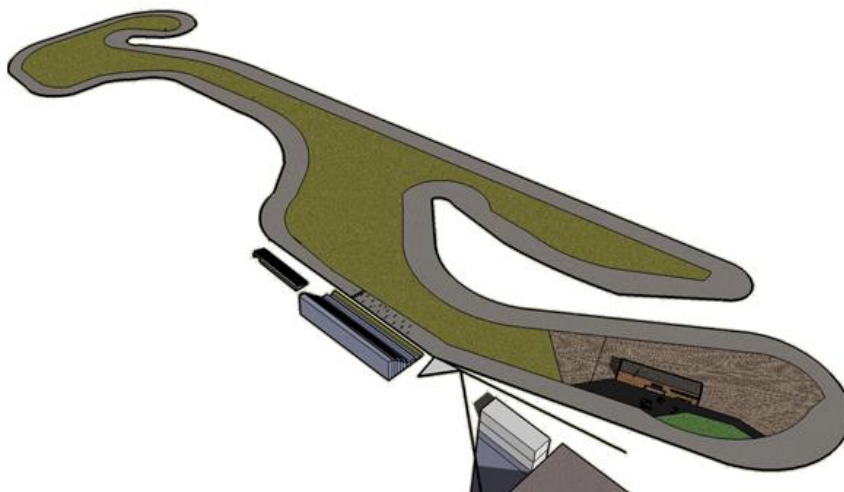


figura 5 - Pista de Nogaró.

¹ Na realidade poderá ser necessário menos, mas ir-se-á “apontar” a meta para 3kms para *worst-case*.

Sistema do ano transacto

No ano transacto [1], no âmbito do projecto realizado obtiveram-se, para o sistema de vídeo, os esquemas a seguir apresentados. É ainda de salientar que este sistema emissor deverá ser instalado no veículo.



figura 6 - Diagrama de blocos 1- sistema emissor.



figura 7 - Diagrama de blocos 2 – sistema emissor.

Numa análise mais detalhada dos diagramas, o modulador2 possuía uma antena embutida e portanto o sistema final, resultava no diagrama de blocos 2, existindo uma versão alternativa do sistema com um modulador mais fraco (-40 dBm de potência de saída) seguido de uma antena monopólo de quarto comprimento de onda, sendo este o diagrama de blocos 1.

Contudo, o sistema nunca conseguiu ultrapassar a distância aproximada de 25 metros de alcance, daí a necessidade de reengenharia do sistema de forma a alcançar os 3 kms.

O sistema de recepção instalado nas boxes seria, nesta configuração, composto por uma televisão ou um computador, com capacidade de aquisição de sinal tv, em conjunto com uma antena monopólo de quarto de comprimento de onda.

Considerações teóricas com novo modelo de radiação

Nos cálculos teóricos efectuados anteriormente, assumiu-se o meio como livre sem perturbações, o que se revelou incorrecto devido à distância a que se encontrava, do chão, a antena do carro. As reflexões provocadas pelo chão, causam um decaimento da qualidade do sinal vs potência, que não é proporcional ao quadrado da distância assumida no trabalho realizado no ano anterior.

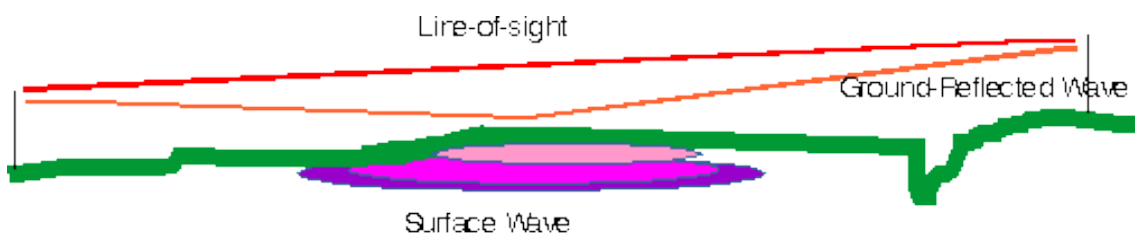


figura 8 - Reflexões provocadas pelo chão[3].

Devido ao baixo comprimento da antena emissora, verificou-se que o modelo adoptado para cálculo da potência necessária para o *link* estava errado. Foi, então, necessário efectuar um novo cálculo [3], [4] com um modelo que tem em consideração, não só a frequência da portadora e a distância, mas também a topografia do terreno (como sendo meio não livre) e a altura das antenas.

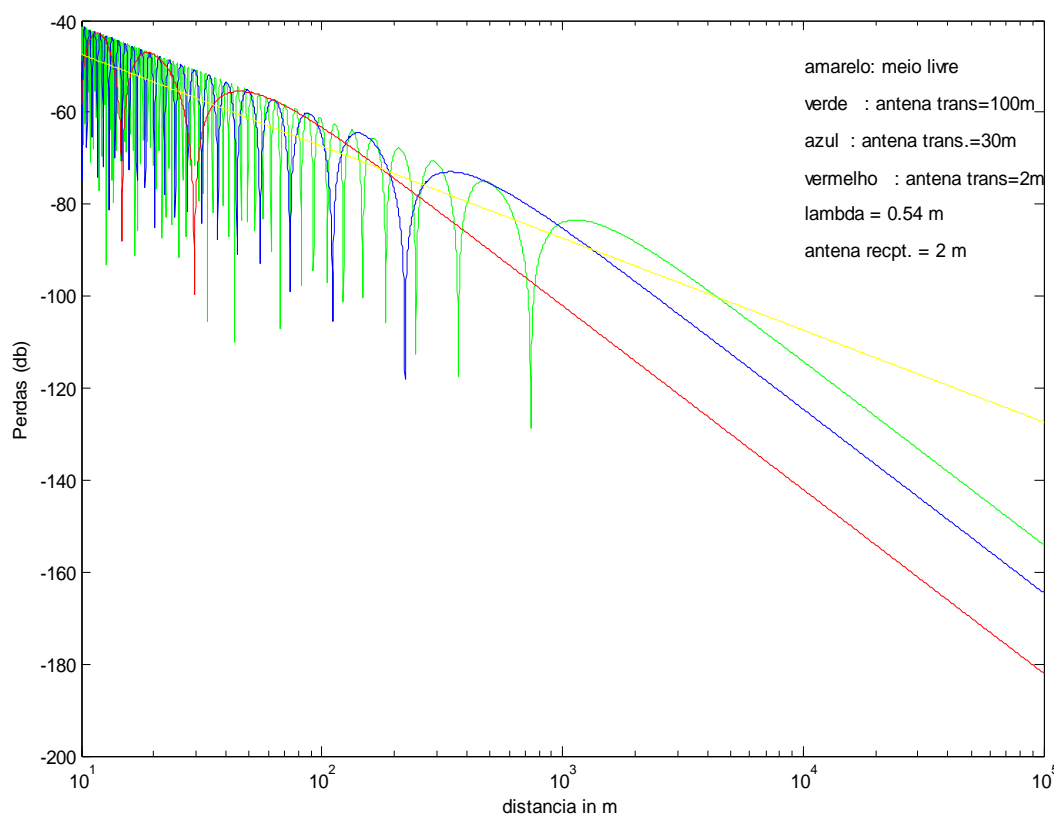


figura 9 - Modelo meio livre vs modelo efeito altura das antenas.

A figura 9 representa a informação de perda em função do comprimento da antena transmissora. É ainda de referir que a antena receptora, nesta simulação, se manteve sempre a uma altura de 2 metros.

Numa análise mais pormenorizada, verifica-se que a onda representada pela cor amarela diz respeito ao meio livre (referente à equação 2.1).

Numa análise comparativa e de acordo com a figura acima apresentada, depreende-se que quer a altura das antenas do solo quer as reflexões no terreno têm uma grande influência na potência transmitida. As arcadas são devido ao cancelamento da onda directa pela onde reflectida no chão.

A grande diferença em relação ao modelo de propagação e radiação anterior é a dependência da potência transmitida com a distância de um factor de cerca de d^4 - devido á distância das antenas emissora e receptora - ao invés do modelo de meio livre que é dependente de um factor de d^2 . Essa é uma diferença significativa em relação ao que inicialmente se pensava ser a potência necessária para o *link* de imagem.

$$P_{RX} = \frac{P_{TX}}{4\pi d^2} \cdot G_e \cdot \frac{\lambda^2}{4\pi} \cdot G_r \quad (2.1)$$

$$P_{RX} = \frac{\lambda^2}{4\pi d^2} \cdot [2 \sin(\frac{2\pi}{\lambda} \cdot \frac{(h_T h_R)}{d})]^2 \cdot P_{TX} \cdot G_E \cdot G_R \quad (2.2)$$

Face a estas conclusões, redefiniu-se a potência necessária para o *link* de vídeo, de forma a se obter uma relação sinal ruído de 28 dB ¹, estando a antena emissora colocada a 2 metros de altura (tal como a receptora). Essa potência é de aproximadamente 5 W ou cerca de 37 dBm, o que é consideravelmente mais em relação ao 1 W, valor assumido no ano anterior.

Deste modo, com base na nova potência de transmissão e com base no esquema do ano anterior, foi necessário adicionar mais blocos funcionais, nomeadamente um amplificador de potência, filtros para eliminação de ruído, bem como uma antena otimizada.

Sistema Actual

O sistema actual, desenvolvido este ano, pode ser dividido em 2 subsistemas:

Sistema de emissão: que será integrado no veículo, sendo portátil e alimentado a(s) bateria(s), permitindo o envio de uma imagem aceitável² para as boxes, por forma a permitir a condução remota do veículo.

Sistema de recepção: instalado nas boxes, que poderá ser directivo e alimentado a partir da rede eléctrica.

Sistema de emissão

Caracterização do sistema anterior

Antes da colocação de um amplificador de potência de forma a obter os 5 W na saída, houve necessidade de otimizar todo o sistema com o intuito de, por um lado, não sofrer perdas de sinal devido a imperfeições significativas do sistema, e por outro obter garantias da fiabilidade do mesmo. Além disso, como já foi referido anteriormente, reviram-se os cálculos teóricos efectuados no ano transacto, visando obter coerência entre os resultados obtidos (no campo e no laboratório) e os esperados teoricamente.

O trabalho iniciou-se pela caracterização do sistema de vídeo, desenvolvido no ano anterior, com a análise dos seus blocos fundamentais, quer em termos de potência como da sua resposta no espectro. Existiam 2 moduladores para serem caracterizados³. Relativamente a um deles foi possível encontrar um *datasheet* no qual a potência de saída era referida como sendo de -40 dBm.

¹ A potência necessária para se ter 40 dB de SNR era de cerca de 60 W devido ao decaimento rápido da curva do modelo adoptado e assumiu-se como aceitável um valor de SNR=28.

² Ver secção Pré-requisitos.

³ Além da antena.

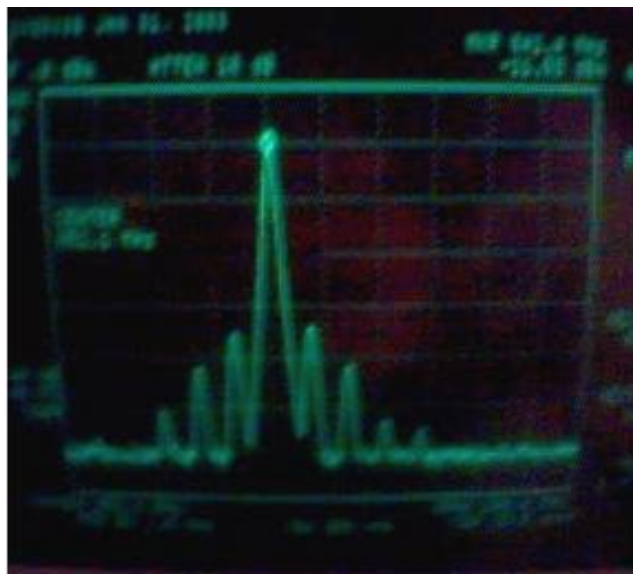


figura 10 - Foto do modulador 2 com pico de -8 dBm e centrado em 570 Mhz.

Em relação ao 2º modulador, este possuía já uma antena incorporada. A análise do seu comportamento recorrendo a um Spectrum analyser¹ indicou um pico de -11 dBm. Verificou-se também que gerava um significativo nível de ruído, resultando na criação de réplicas em diversas frequências ao longo do espectro. Por este facto, foi necessário, para este caso, criar-se um filtro passa-banda para UHF.

A figura 10, embora não seja muito nítida, ilustra a forma de onda do sinal de UHF, após modulado.

Num momento posterior, caracterizou-se também a antena monopolo criada no ano anterior, por forma a conhecer que frequências de trabalho apresentam melhor VSWR², com vista a otimizar a maior potência transmitida para a saída, maximizando assim o rendimento do sistema.

¹ Daqui em diante, todas as referências feitas neste capítulo de testes de laboratório dizem respeito a testes efectuados com auxílio de Spectrum analyser.

² Voltage Standing Wave Ratio

Monópolo - antena ano (06/07)

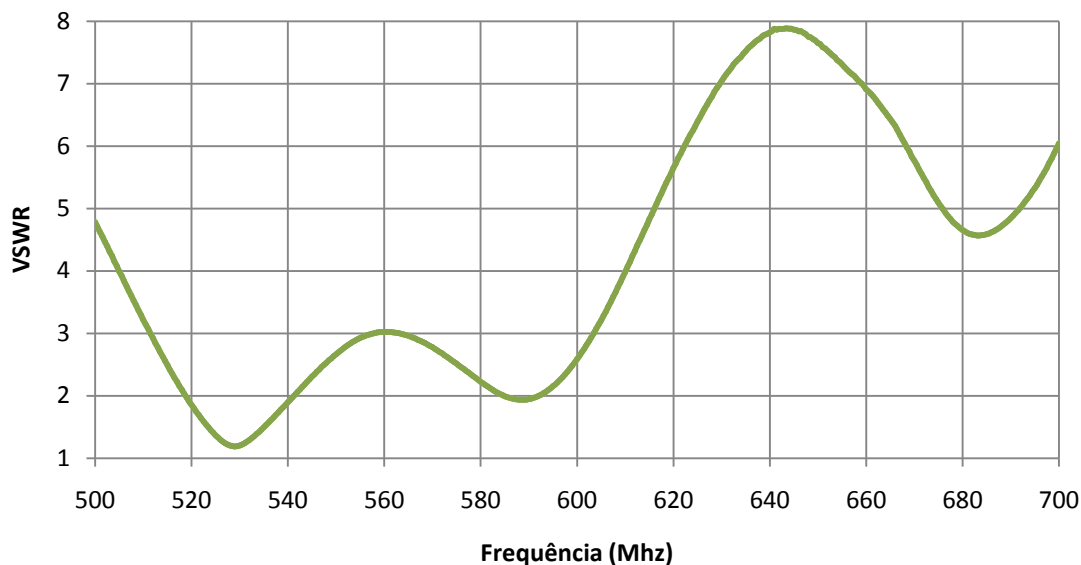


figura 11 - Antena monopolo – do ano transacto.

Esta antena apresenta algumas deficiências, entre as quais um VSWR elevado para as frequências de trabalho, sendo a excepção à volta de 530 Mhz. Outra das particularidades desta antena é que não apresenta uma curva de VSWR considerada normal para um monopolo¹ de quarto de comprimento de onda. O ideal seria um VSWR baixo, sendo os valores óptimos entre 1 e 2 para as frequências que se irão utilizar para transmitir vídeo.

Desta forma perde-se alguma potência na saída do sistema que poderia ser transmitida.

Pré-amplificador

Face à verificação de que o sinal proveniente do modulador tinha amplitude insuficiente para alimentar o amplificador de potência do sinal do modulador, foi necessário colocar um andar de pré-amplificação entre estes dois dispositivos.

Numa primeira fase, colocou-se um amplificador de recepção de antena comercial, com um ganho de 20 dB [5], trocando apenas a entrada de antena pela do modulador² e a saída para televisão pela de antena, pois o pré-amplificador é recíproco. Salienta-se ainda que o amplificador era alimentado, não pela rede eléctrica, mas sim a partir de uma bateria.

¹ Um monopolo tem uma curva de VSWR que apresenta uma concavidade em relação ao eixo das frequências no caso do gráfico da figura 19.

² A única alteração feita no modulador consistiu em retirar a antena incorporada e colocar um conector BNC fêmea.

O amplificador comercial possuía uma limitação em amplitude do sinal de saída, e consequentemente, nunca se conseguiu o *boost* necessário no sinal que era esperado. Todavia, apresentam-se de seguida os resultados obtidos.

Resultados obtidos:

Em campo [6]:

	Modulador 1	Modulador 2	M1+ amp	M2+amp
Distância (m)	0.5	25	25	110



figura 12 - Distância M2 + amplificador (IEETA na imagem).

Em laboratório:

	M1	M2	M2+ amp
Potência (dBm)	-40	-11	-6

Embora no *datasheet* [5] refira que a saída do sinal é superior a 106 dB μ V, verifica-se que o sistema satura. Fazendo a conversão de dB μ V para dBm, 106 dB μ V equivale a uma saída de -6 dBm, daí a passagem de -11 dBm somente para -6dBm e não para mais.

Deste modo alterou-se o pré-amplificador para um amplificador GaAs¹, tendo este somente um ganho de 10 dB com banda de 0-2Ghz². Embora com um ganho inferior, este amplificador, não apresenta a limitação na saída evidenciada pelo amplificador comercial.

Testes efectuados com o amplificador GaAs:

Em laboratório:

Pico de potência de 5.4dBm a 570Mhz.

Em campo:

Distância de cerca de 250 metros (ver figura 13).

¹ Fornecido pelo Professor Violas.

² Devido à LB do amplificador será necessário utilizar um filtro passa-banda para UHF.



figura 13 - Modulador 2 + amp. GaAs.

A simulação realizada em matlab com o modelo de propagação seria de esperar que se conseguisse quase 400 metros de alcance, caso a potência à saída do amplificador fosse toda radiada, logo o sistema necessitaria de ser otimizado antes da colocação do amplificador de potência. Colocou-se então um filtro UHF para eliminar espúrias criadas pelo modulador, e que fazem perder potência, e também uma antena melhorada e otimizada para as frequências de utilização.

Filtro UHF

Para eliminar as espúrias que se observaram em laboratório, colocou-se um filtro passa-banda para UHF que, neste caso, percorre a gama de 440 – 800 Mhz. No início pensou-se fazer o filtro em *microstrip* [7]. Porém, como as linhas são de $\lambda/4$ e como trabalhamos num canal de UHF a 570 Mhz, as linhas teriam um comprimento de $0.52/4=0.13$ m. Utilizando transformadas de *kuroda* para alterar a bobine, que é um stub em série, e passá-lo para paralelo, obtêm-se um comprimento do filtro de $3*0.13 \approx 39$ cm, se considerarmos um filtro de 1ª ordem. O comprimento do filtro seria assim muito grande, tendo-se optado por componentes discretos SMD¹ para o referido filtro.

¹ surface-mount devices.

Simulou-se, em Microwave Office, o circuito seguinte:

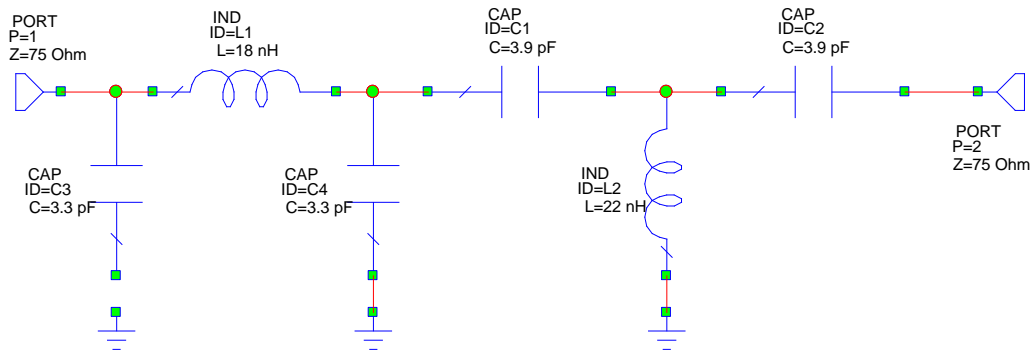


figura 14 - Esquema do filtro passa-banda.

O filtro consiste num andar em π , seguido de um andar em T, que atenuam as frequências abaixo de 440 Mhz e acima de 800 Mhz. Um filtro com mais andares, por forma a fazer subir a sua ordem, seria possível. Todavia, os resultados apresentaram-se satisfatórios para o fim em vista, permitindo assim a existência de um filtro mais compacto e com menos componentes.

Obtiveram-se, assim, os seguintes resultados em simulação:

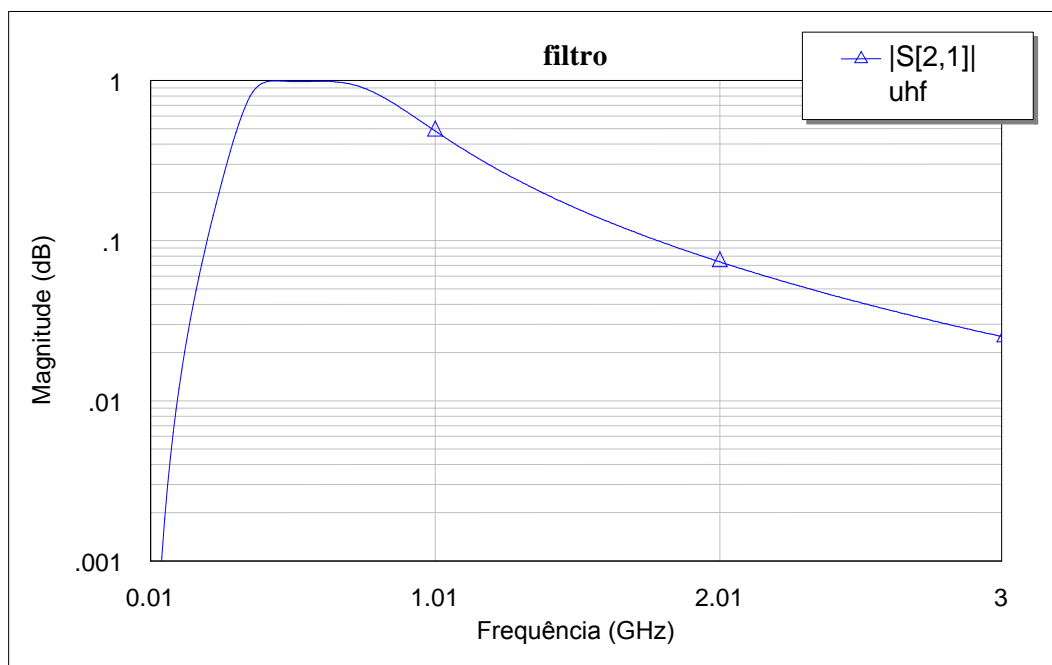


figura 15 - Simulação do filtro UHF.

Em laboratório, após a criação física do filtro com os componentes existentes na figura 14 obtiveram-se os resultados patentes na figura 16:

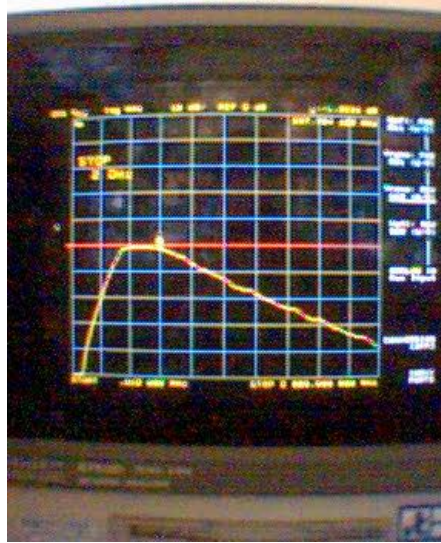


figura 16 - Filtro UHF, resultado em laboratório.

Embora a imagem de laboratório não esteja com a resolução que seria desejável, verificou-se que os resultados obtidos eram em muito idênticos aos da simulação. A banda passante vai dos 450 Mhz aos 790 Mhz, sendo as diferenças em relação ao circuito simulado resultado da imprecisão associada aos componentes SMD. Os resultados da simulação vs laboratório são, contudo, muito satisfatórios.

Somente para efeitos de registo, apresentam-se as imagens que dizem respeito ao aspecto do filtro passa-banda.

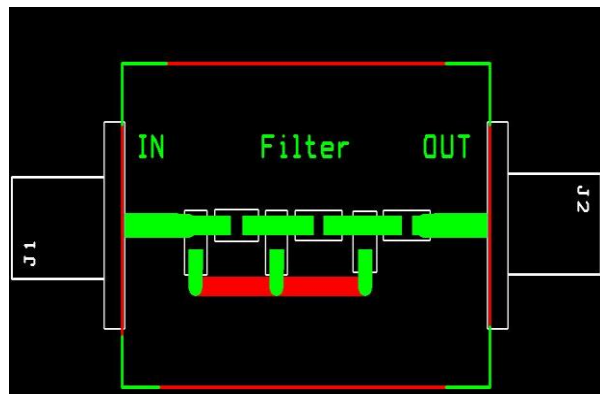


figura 17 - Filtro UHF.

Antena

Tal como foi referido anteriormente, a antena não se encontrava otimizada, razão pela qual se estava a perder potência que poderia ser radiada com a construção de uma antena com o comprimento correcto.

Neste sentido, foi desenvolvida uma nova antena que, numa 1ª fase, foi simulada e testada em laboratório de modo a conseguir radiar o máximo de potência possível.

A antena emissora tem de ser um monopólo de um quarto de comprimento de onda para radiar em todas as direcções [6] pois o veículo andarà numa pista, o que não permite directividade.

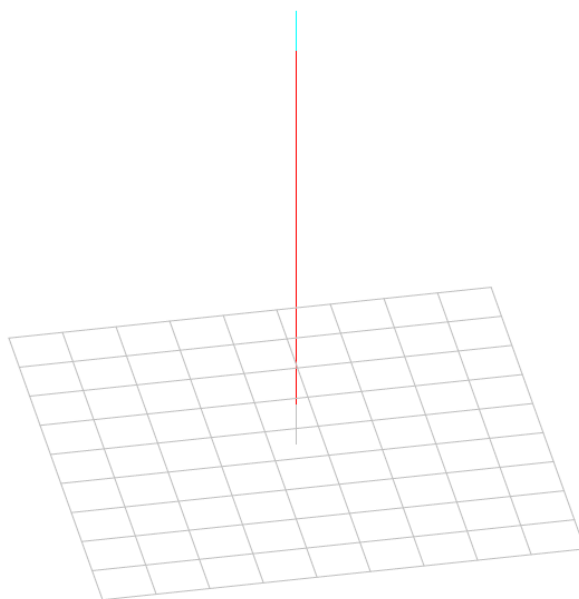


figura 18 - Antena monopolo simulada.

A nova antena terá um plano de massa e será em alumínio (por ser mais fácil de trabalhar). O fio da antena, na imagem representado a vermelho, será em cobre.

Através de uma ferramenta para simulação e projecto de antenas, baseada em *Numerical Eletromagnetic Code (NEC)*¹, projectou-se assim uma antena que fosse otimizada para transmitir no canal 31, que começa em 570 Mhz.

¹ Um agradecimento especial ao Professor Rocha Pereira pela ajuda prestada.

Os resultados obtidos, através da simulação da nova antena, são os seguintes:

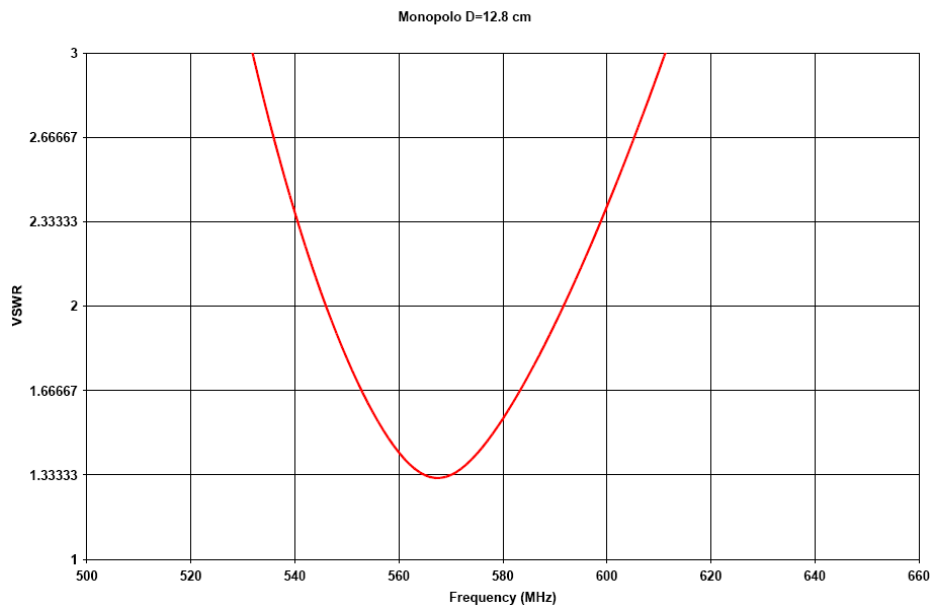


figura 19 - Gráfico do monopólo simulado.

De acordo com diversas simulações efectuadas, verificou-se que quanto menor for o comprimento da antena, mais o diagrama de VSWR¹ se deslocava para a direita, chegando ao valor teórico de 12.8 centímetros para o comprimento da antena. Tal facto é visível através do gráfico apresentado na figura 19.

Desta forma construiu-se então a antena em alumínio e, com o *Spectrum Analyser*, optimizou-se para os 570 Mhz, obtendo-se o seguinte diagrama de VSWR para a antena real.

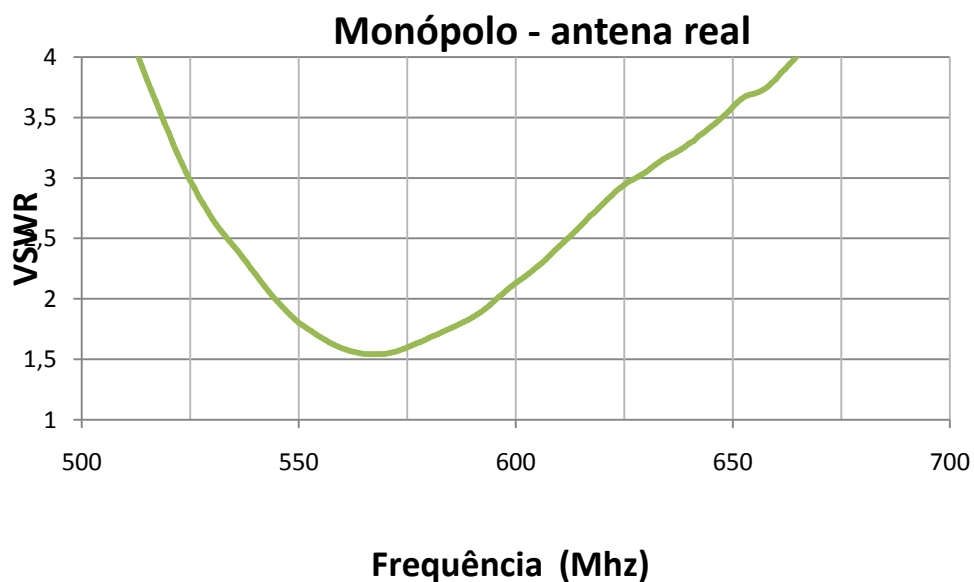


figura 20 -Gráfico do monopolo real.

¹É de notar que o VSWR vai de 1 a ∞ , sendo os valores que queremos obter o mais próximo de 1 para optimização.

A antena apresenta uma boa resposta para a frequência pretendida, possibilitando, assim, uma maior potência radiada no canal 31 de UHF. É possível ser utilizada também noutros canais, garantindo uma maior potência radiada relativamente à anterior.

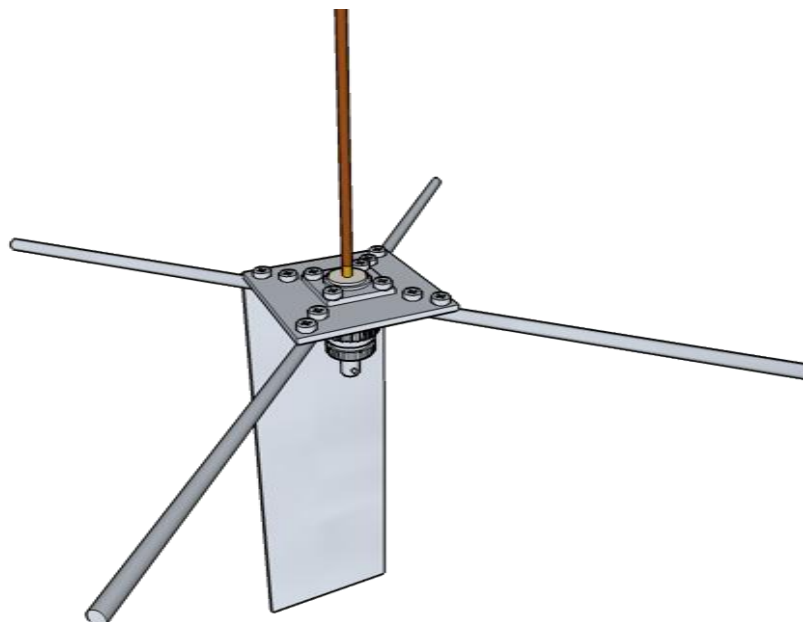


figura 21 – Nova antena.

Resultados auferidos nos testes de campo:

A distância obtida é sensivelmente a mesma que se obtinha com o modelo anterior contudo, conseguindo-se agora entre 10 e 20 metros a mais de alcance. Aponta-se como limitação destes testes, o local onde foram feitas as medições, uma vez que, a 220 metros existem árvores e, porventura, veículos estacionados. Com estas condições, o sinal perde-se rapidamente porém, as diferenças em relação à anterior são notórias. Uma vez que a qualidade de imagem é significativamente melhor, considera-se que esta antena evidencia uma maior potência radiada.

No que diz respeito à introdução do filtro, não se verificou nenhuma vantagem em termos de potência. Ainda assim, será aconselhável a introdução deste módulo no sistema final, pois o modulador M2 introduz bastantes esporas, em zonas do espectro que não queremos transmitir. Tal situação será minimizada pela utilização do filtro que, por sua vez, eliminará os harmónicos introduzidos pelo modulador.

Amplificador de Potência

Logo que o sinal já se encontrava com alguma potência, uma vez que as esporas criadas pelo modulador tinham sido eliminadas pelo filtro e a antena apresentava um

elevado grau de fiabilidade, este era um sistema que apresentava uma boa resposta mas com pouco alcance.

Uma vez que, teoricamente, se tinha uma boa noção da potência necessária para o *link* de vídeo com as boxes, compraram-se transístores de potência de 5W [8] para colocar no final da cadeia, de forma a estabelecer uma ligação directa com a antena.

O amplificador foi desenhado para a frequência de 500Mhz com *trimmers*, com o intuito de conseguir uma melhor adaptação da malha de entrada e de saída, induzindo a obtenção máxima de potência de saída.

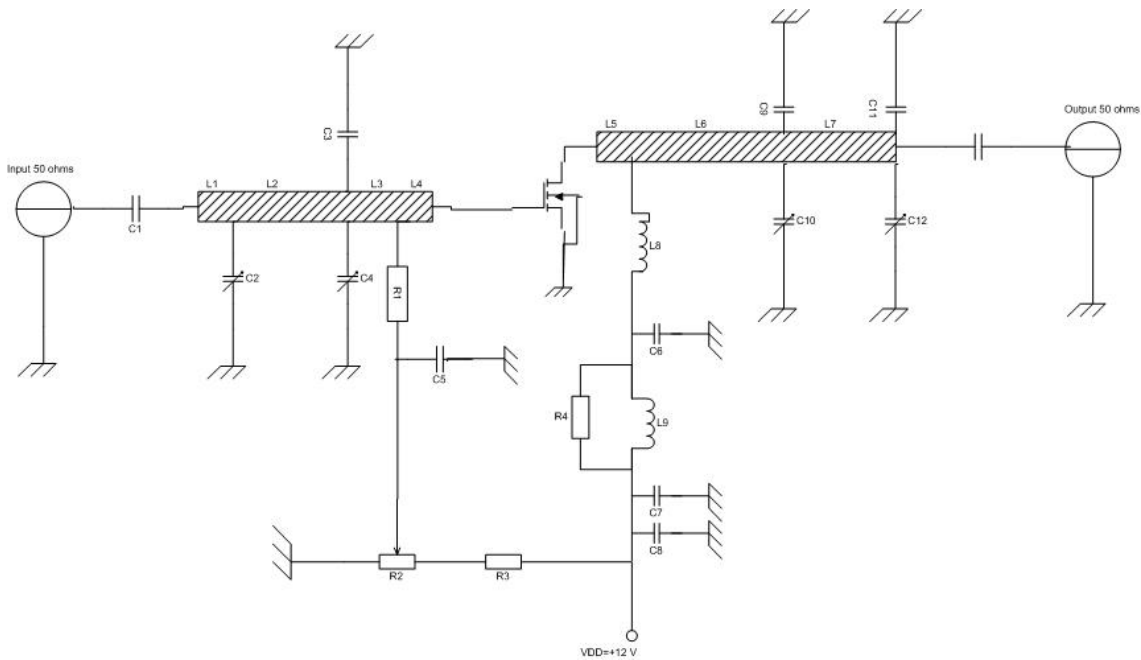


figura 22 - Circuito do amplificador de potência.

De seguida, apresenta-se o *layout* da placa. O *layout* foi feito em autocad, com a ajuda de um membro do IT, Paulo Gonçalves. Foi construída a placa dissipadora em alumínio, que servirá também de suporte ao transístor. Esta placa foi feita com o auxílio do professor Bernardo Cunha.

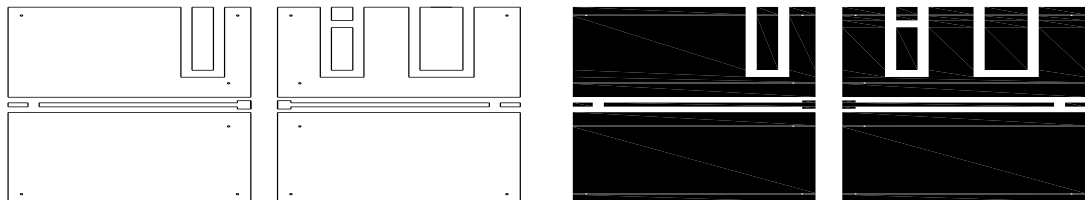


figura 23 -Amplificador de potência.

As imagens apresentadas em cima dizem respeito à placa e ao negativo utilizado para o circuito impresso. Como é possível observar, a placa é cortada a meio, de forma a permitir a colocação do transistor de potência. Ambas as placas possuem *trimmers* para melhorar a adaptação na malha de entrada. Na parte superior das placas encontram-se as alimentações do circuito.

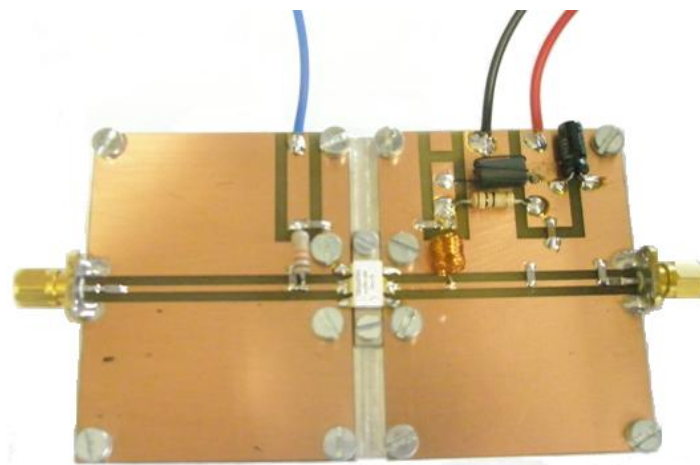


figura 24 - Aspecto real - amplificador de potência.

Resultados Finais

O sistema desenvolvido, não chegou a ser completamente implementado. No entanto, devido ao sinal disponibilizado pelo amplificador GaAs, foi necessário colocar um amplificador de 0.5 W¹ de forma a aumentar a força do sinal. Além disso o conector utilizado na saída do modulador foi alterado devido a não estar bem fixo e assim introduzir ruído e atenuar o sinal.

O esquema seguinte ilustra o sistema final para atingir os objectivos propostos inicialmente.



figura 25 - Estado final do sistema de emissão.

O sistema de 5W não se conseguiu no âmbito desta tese, colocar em funcionamento, necessitando somente da integração de um dissipador no amplificador e recorrer a um *Spectrum Analyser* para fazer os ajustes devidos com os *trimmers*.

O sistema ficou portanto a funcionar até ao amplificador de 0.5W apresentando-se de seguida os resultados em laboratório do sistema.

¹ Disponibilizado pelo Professor Violas

	Amp. GaAs	Amp. GaAs +amp. 0.5W
Potência á entrada da antena (dBm)	8,5	21

Tabela 1 - Resultados em laboratório

O sistema apresenta já um sinal elevado, embora ainda longe dos 37 dBm. Porém a introdução do amplificador de 5W no sistema elevará a potência do sinal, colocando-o perto dos 37dBm.

Sistema de Recepção



figura 26 - Sistema de recepção.

O sistema de recepção de imagens, localizado nas boxes, consiste numa televisão ligada a uma antena. Esta foi alterada de monopólo para uma antena comercial mais directiva¹ e com elementos activos que permitem um maior ganho. Testes efectuados este ano e também no ano anterior mostraram que o computador introduz um atraso de cerca de meio segundo, não permitindo, por isso, um visionamento em tempo-real das imagens obtidas. Esta situação deve-se ao processamento feito pela placa de tv externa do portátil que introduz o atraso que se verificou no visionamento.

¹ O sistema de recepção será colocado nas boxes, podendo ser muito mais directivo que a do sistema de emissão.

Conclusões

Do ponto de vista funcional, o sistema de vídeo sofreu diversos melhoramentos, relativamente ao ano anterior. A alteração do diagrama de blocos funcionais e a adaptação do modelo de propagação e radiação, mostrou-se mais coerente com os resultados observados experimentalmente.

Com o novo modelo, surgiu também a necessidade de se obter uma maior potência na saída em relação ao que se pensava inicialmente, tornando o sistema mais complexo.

A principal dificuldade encontrada no desenvolvimento deste sistema prende-se, essencialmente, com as informações provenientes do trabalho desenvolvido no ano transacto. O volume de informação, em termos de caracterização, era escasso, pelo que foi necessária uma melhor descrição do sistema anterior, bem como do actual, de forma a conseguir justificar decisões tomadas e caminhos seguidos. Esta metodologia revela-se bastante eficaz na medida em que, se for necessário fazer alterações no sistema, poder-se-á fazê-lo com bastante informação disponível.

Do ponto de vista académico, foi-me possível clarificar as noções de propagação e radiação de ondas, de criação de sistemas funcionais ao nível de antenas, filtros, pré-amplificadores e amplificadores de potência, entre outros.

Em relação a trabalho futuro, falta a integração do amplificador de 5W no sistema, estando este já criado faltando fazer os ajustes necessários e testá-lo no sistema de emissão.

Capítulo 3

Alterações efectuadas no volante

Neste capítulo dar-se-á ênfase às alterações efectuadas no sistema volante, quer do ponto de vista funcional, de forma a melhorar a resposta do volante, quer do ponto de vista estrutural, possibilitando a sua integração completa no sistema global ICARO.

Introdução

O volante é constituído pelo módulo de comunicação e leitura dos comandos fornecidos pelo condutor e, como se verificará mais adiante, conterà também o *palmtop/PDA*, responsável pelas informações que serão fornecidas ao condutor¹.

Em ambos os modos de condução, remota e normal, o sistema volante permite ao piloto ligar e desligar o motor, buzinar, actuar sobre o acelerador e o travão (este somente quando o sistema se encontra em modo de operação remota). Da mesma forma também é possível ao condutor do veículo aceder aos vários ecrãs do *palmtop/PDA*, continuando a controlar o carro, quer no modo de condução remota quer quando o volante se encontra conectado ao veio da direcção.

Para além dos 4 botões do volante anteriormente referidos, é de notar também a existência de um potenciómetro, responsável pelo fornecimento do valor da aceleração desejada a empregar no veículo, bem como um sensor de carga, que accionará o dispositivo relativo ao travão, quando em condução remota.



figura 27 - Aspecto geral do volante [1].

¹O sistema *palmtop/PDA* será referido num capítulo posterior, uma vez que o objecto de estudo aqui referido é o módulo volante, a electrónica e código que lhe estão associados.

Alterações efectuadas - causas:

Tal como já foi referido anteriormente, o volante possui um acelerómetro que, ao fornecer os dados das componentes dos 3 eixos – no referencial cartesiano – permite o cálculo do ângulo que o volante faz em relação ao veio da direcção. Porém, quando aplicadas forças horizontais e/ou verticais (e não em torno de um eixo), o acelerómetro fornece dados que resultam em ângulos não verdadeiros.

Abordou-se, então, o problema, testando 2 possibilidades de forma a atenuar a situação acima descrita. As hipóteses consideradas são as seguintes:

- o cálculo da aceleração gravítica, através da expressão $x^2 + y^2 + z^2 < g^2 + threshold$. Caso o resultado do cálculo do 1º membro da expressão seja superior, relativamente ao 2º membro, significa que se está a criar aceleração, segundo uma ou mais das componentes;
- a adição de um módulo de giroscópio no veio da direcção, perpendicular à placa que contém a electrónica do volante. Deste modo “validam-se” os valores do acelerómetro, uma vez que o giroscópio apenas fornece valores de tensão para forças radiais.

Cálculo da aceleração:

Adoptando como metodologia a expressão de $x^2 + y^2 + z^2 > g^2 + threshold$, caso se verifique esta condição, mantém-se o valor do ângulo de direcção anteriormente calculado.

O cálculo da aceleração provou ser eficaz para acelerações elevadas, observando-se que o volante, para acelerações superiores a $g + threshold$, mantinha o valor anterior do ângulo de direcção.

Contudo, para acelerações muito baixas (ou mesmo nulas), segundo uma ou mais das componentes que não fosse o eixo de direcção, esta técnica não é capaz de corrigir o problema. Tal situação deve-se ao facto do deslocamento ser feito muito lentamente, não criando uma aceleração significativa em relação à que obteríamos caso estivesse em repouso ou a mover-se segundo um eixo.

Criação do módulo Giroscópio:

O giroscópio é responsável pelo fornecimento de valores de tensão, proporcionais a forças radiais. Para tal, o giroscópio [9] teria de ser colocado segundo o eixo do veio de direcção, de forma a ser possível obter os valores mais correctos. O giroscópio, em conjunto com o acelerómetro, permitiria alcançar valores bastante correctos de direcção - o esquema eléctrico e o *layout* da placa encontram-se no anexo C.

Em relação ao código [10] [11], o ganho do amplificador que se encontra em frente do giroscópio tem o valor de 5. Tendo o giroscópio uma sensibilidade de 0.62mV/°/s, obtem-se um ganho de 3.1 mV/°/S.

A ADC é de 10 bits possuindo um VDD = 5V, logo:

$$1 \text{ bit} = \frac{5}{1024} \cong 4.88 \text{ }^{\circ}/\text{s/bit}$$

O período de tempo em que vai ser executado o excerto de código, referente ao giroscópio, é de 0.05 s, o que corresponde a 20 Hz.

Como exemplo, observe-se:

Valor 0 corresponde a 512 da ADC do microcontrolador. Imaginemos que a ADC devolve o valor 520, então:

$$520 - 512 = 8 * 4.88 = 39.04 \text{ (}^{\circ}/\text{s)} * t = 39.04 * 0.05 = 1.95^{\circ}$$

O que significa que, naquele intervalo de tempo, ocorreu uma variação de 1.95 graus.

Com base nesta metodologia criou-se, então, o código que serviu de base no microcontrolador. Ainda assim nunca foi possível obter valores satisfatórios - com o giroscópio - que justificassem a introdução deste na versão final do volante.

Com o intuito de melhor avaliar a *performance* do giroscópio, fizeram-se dois testes de forma a entender o que estaria a condicionar o seu desempenho. Assim sendo:

1º teste

Através de um módulo bluetooth¹ com um giroscópio, foi possível testar o funcionamento deste com velocidade constante. Para isso, colocou-se o dispositivo sem fios sobre o eixo de um motor DC, sempre alimentado à mesma tensão.

Os resultados obtidos foram os seguintes:

¹ Uma palavra de apreço ao Eng. David Ribeiro por ter fornecido o dispositivo sem fios.

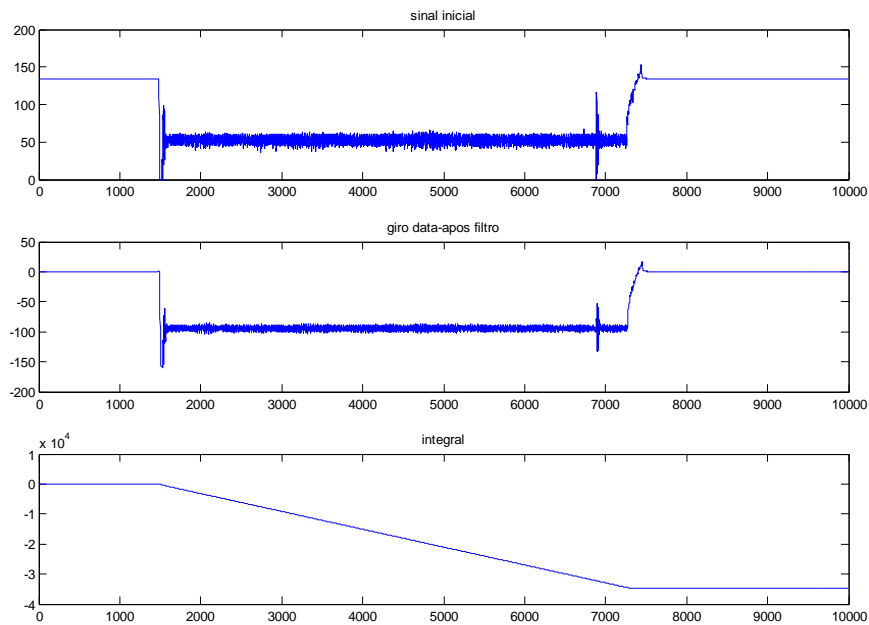


figura 28 - Sinal do giroscópio a velocidade constante.

Analisando a figura 28 verifica-se que, inicialmente, o dispositivo se encontrava parado. À *posteriori*, o motor foi ligado, deixando-se o dispositivo a girar em torno do eixo do motor durante algum tempo, até se parar o motor DC. Verifica-se, então, que enquanto o motor gira sempre no mesmo sentido, o valor do integral do giroscópio aumenta linearmente (não se colocou nenhuma condição para quando chegasse aos 360 graus, pois o mais importante era verificar os valores fornecidos pelo giroscópio), incrementando a variável de posição existente. Prova-se, assim, que o giroscópio apresenta valores proporcionais à velocidade.

2º teste

De seguida efectuou-se um novo teste, utilizando o volante existente e o giroscópio criado, ligado à electrónica do volante. O giroscópio estava colocado no veio da direcção e executou-se um movimento pré-definido com o volante.

Esse movimento foi executado da seguinte forma: em 1º lugar rodou-se o volante para a direita, a 90 graus. De seguida, efectuou-se o mesmo movimento, mas para a esquerda e novamente para o centro, voltando à posição inicial. Num 3º movimento, o volante foi rodado para a esquerda, a 90 graus e, finalmente, para a direita 90 graus, em direcção ao centro.

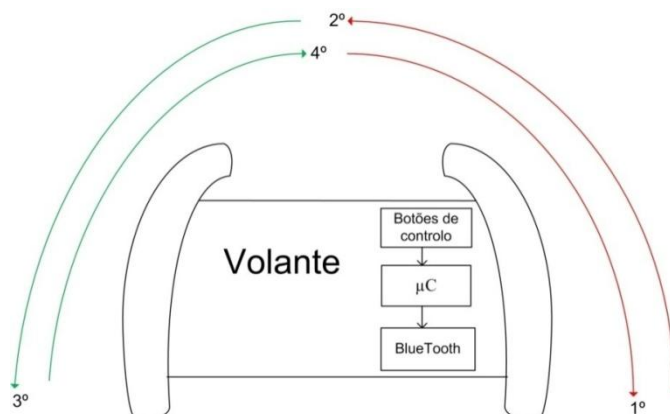


figura 29 - Movimento realizado.

Com os movimentos enunciados e exemplificados com recurso à figura 29, obtiveram-se os seguintes gráficos:

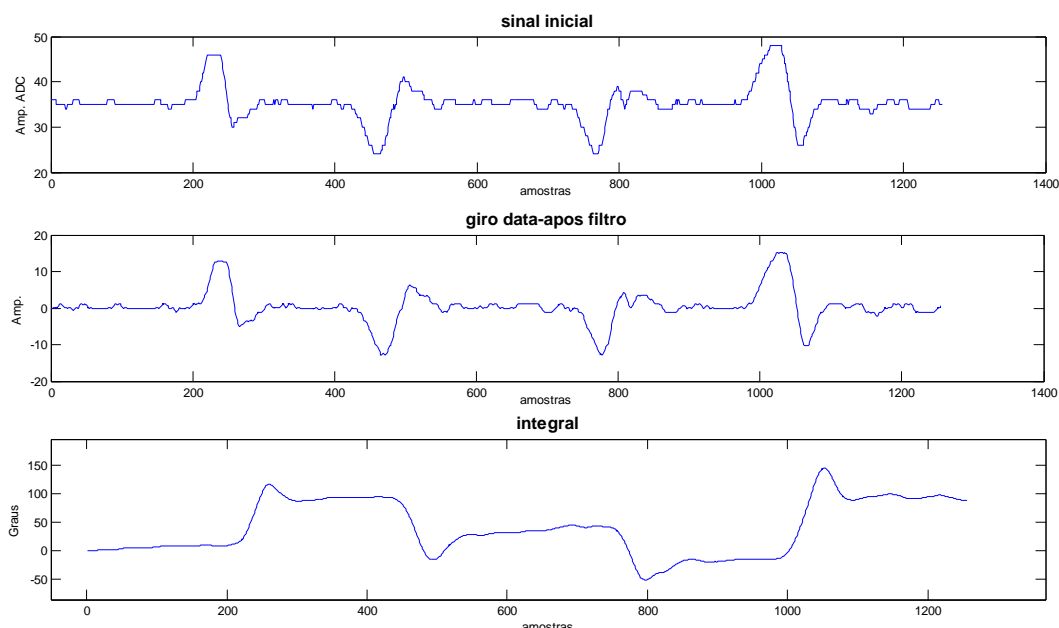


figura 30 - Sinal proveniente do giroscópio, movimento conhecido à partida.

O 3º gráfico corresponde ao sinal integrado, proveniente do giroscópio, sendo que o 1º gráfico diz respeito ao sinal original. Já o 2º é o resultado do sinal após filtragem.

Percebe-se claramente em que situações o volante foi girado para a esquerda e para a direita. Como era de prever, existem 4 picos principais relativos à rotação do volante, nos 4 movimentos exemplificados. As 2 ondas do meio apresentam características semelhantes, tal como as dos extremos.

Embora, no 3º gráfico seja necessário um factor de escala, observa-se um efeito comum nestes dispositivos - o *drift*.

Quando em repouso, numa determinada posição, o dispositivo vai incrementando o seu valor, não permanecendo continuamente no mesmo sítio. Tal situação é visível através do 3º gráfico. Uma forma de minimizar este efeito é utilizando um acelerómetro (como o existente no volante), de forma a rectificar o valor absoluto fornecido pelo giroscópio ao final de algumas amostras, que se ache significativo.

Com esta técnica, ainda assim, verifica-se que o giroscópio continua a dar valores muito errados. Como tal, esta hipótese ficou “suspensa”.

É de notar que existem giroscópios com melhor compensação de *drift*, ADCs internas, bem como melhor sensibilidade do que o giroscópio utilizado. Estes utilizam comunicação SPI¹ para a ligação sendo porém, muito mais dispendiosos.

Por outro lado, o volante, mesmo em condução remota, estará fixado a uma mesa, não permitindo, assim, efectuar os movimentos referidos anteriormente – movimentos não centrífugos - que originam esses erros.

Alteração na taxa de amostragem

A aceleração, travão e direcção eram amostrados a uma frequência de 20 Hz, o que, para os movimentos em questão, é bastante elevado. Deste modo, reduziu-se essa amostragem, alterando os valores do filtro FIR para os 6 Hz.

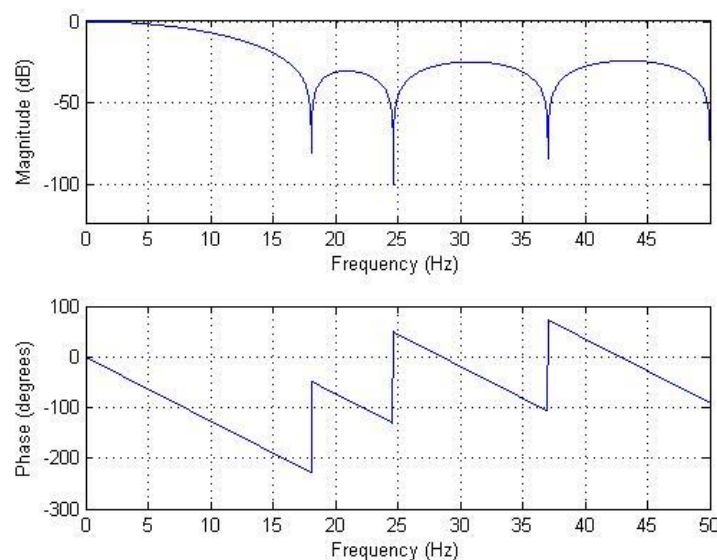


figura 31 - Gráfico do Filtro FIR.

```
>> F=0:0.01:50;  
>> freqz (FIR1(7,3/20,'low',chebwin(7+1,20)),1,F,100)
```

Os coeficientes dos filtros FIR, alterados para uma amostragem de 6 Hz, têm como consequência menor sensibilidade e, consequentemente, uma maior robustez a variações

¹ Serial Peripheral Interface.

mínimas na aceleração, travagem e direcção. Apesar disso, o ponto crítico do sistema é a direcção como foi verificado experimentalmente.

Alteração no monitor ICARO

O monitor ICARO é um programa que permite monitorizar todos os sistemas do veículo das boxes, além do envio de comandos para o veículo. No referido monitor apenas se adicionou código para que, ao pressionar do botão `screen_change` do volante, fosse visível no monitor ICARO. Assim, torna-se possível interagir o volante com o monitor, observando todos os botões do volante a funcionar.

Alteração na tabela das prioridades [1]:

No ano transacto, o *palmtop*/PDA desempenhava um papel crucial na criação e manutenção da rede, pois servia de *gateway* entre volante e módulo de controlo. Este ano, foi alterada¹ essa técnica, visando tornar o sistema mais fiável e rápido. Assim sendo, a tabela de ligação do volante e de prioridades foi alterada.

De acordo com o esquema seguinte – figura 32 - a ligação a amarelo diz respeito às ligações sem fios e a verde às ligações pelo barramento CAN.

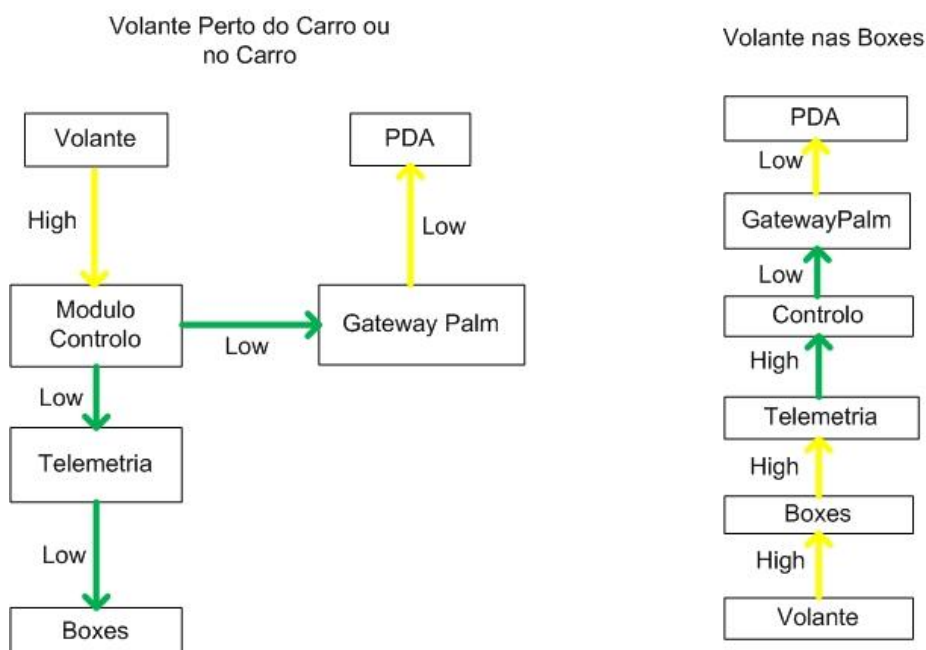


figura 32 - Prioridades das ligações.

Como se pode observar, as ligações relacionadas com envio de tramas para controlo são todas de alta prioridade, sendo as de monitorização e de visualização ao piloto de baixa prioridade.

¹ Num capítulo posterior (capítulo 5) dar-se-á mais ênfase à metodologia adoptada.

Nesta nova configuração, o *palmtop/PDA* já não desempenha um papel crucial na rede, passando apenas a ser um elemento terminal desta, não deixando de ser responsável pelo *logging* dos dados e a consequente visualização por parte do piloto. Quando no veículo, o volante deixou de se ligar ao *palmtop/PDA*, para se ligar directamente ao módulo de controlo.

Alteração do comando - activação de controlo remoto

É possível teleoperar o veículo e activar a condução remota através de 2 formas:

- Perto ou dentro do veículo;
- Enviando os comandos da boxe.

O método de activação quando perto do veículo consistia em colocar o volante “deitado”, formando um ângulo de 90 graus, em relação ao tronco do piloto, colocando-se novamente na posição “normal” de condução. Esta técnica permitia aproveitar a coordenada Z do acelerómetro porém, era de difícil execução, por isso a activação da teleoperação perto do carro é, agora, possível pressionando o botão start e, de seguida, o botão screen, com um determinado intervalo de tempo máximo entre estes 2 botões.

Conclusões

O volante, embora já estivesse bastante funcional, apresentava alguns problemas. Como tal foi necessário introduzir algumas modificações e/ou ajustes, visando a sua funcionalidade plena no sistema global ICARO.

Não foi possível colocar o giroscópio totalmente operacional e, como tal, não foi incluído na versão do volante. Porém, por sugestão do Eng. David Ribeiro poder-se-á experimentar um outro tipo de giroscópio (ADIS16250 da Analog Device), uma vez que possui maior sensibilidade e ADC interna para o efeito pretendido.

Capítulo 4

Projecto e desenvolvimento de um sistema de actuação controlado por um microcontrolador capaz de conduzir e travar o veículo

No desenvolvimento deste capítulo serão analisadas as alterações efectuadas no sistema Módulo de controlo. Discutir-se-á, então, a integração do sistema electrónico com a mecânica do veículo, bem como os diferentes testes efectuados e as alterações na arquitectura onde se enquadra o módulo de controlo/GatewayPalm.

Introdução ao sistema

O sistema desenvolvido no estado do ano transacto, apresenta electrónica que é responsável pelo reencaminhamento da CAN para o *palmtop/PDA* e do reencaminhamento da informação do volante para a CAN do veículo.

Apresentava, também, a possibilidade de controlo da direcção e do travão - quando em condução remota. Esse controlo era feito com o auxílio de 2 motores DC, onde era introduzida uma onda PWM na sua alimentação, de forma a colocá-los no lugar desejado, quer para a direcção quer para o travão.

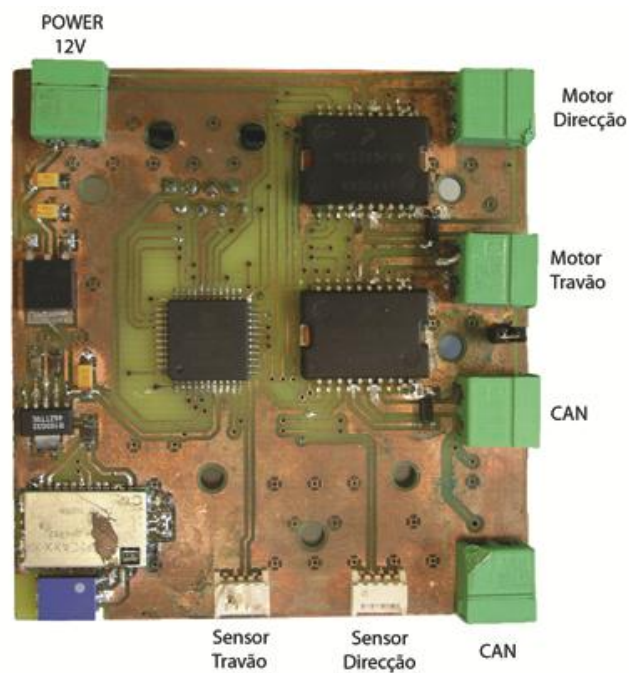


figura 33 - PCB do módulo de controlo. [1]

Anteriormente, existia uma malha de realimentação, que possibilitava conhecer o local onde se encontrava o motor em questão e, deste modo, fazer o seu controlo em malha fechada.

Assim, neste capítulo analisaremos os itens que estavam por concluir/realizar, nomeadamente o travão, além da integração da electrónica no veículo e alterações efectuadas, visando o correcto funcionamento de todo o sistema, quer em condução normal quer em condução teleoperada.

Travão

Do ponto de vista material [1], o travão não se encontrava criado havendo, contudo, modelos relativos à sua estrutura física e indicadores da forma como este se iria comportar fisicamente.

A ideia original assemelhava-se a um pé no pedal (de forma a empurrar o travão existente no veículo). O motor de rotação estaria, então, ligado a um sem-fim que transformaria a força rotacional em força de translação, através de um guia.

Assim sendo, a força rotacional seria traduzida num avanço ou recuo do pedal, conforme desejado pelo piloto, e a força que este empregasse no sensor do travão existente no volante.

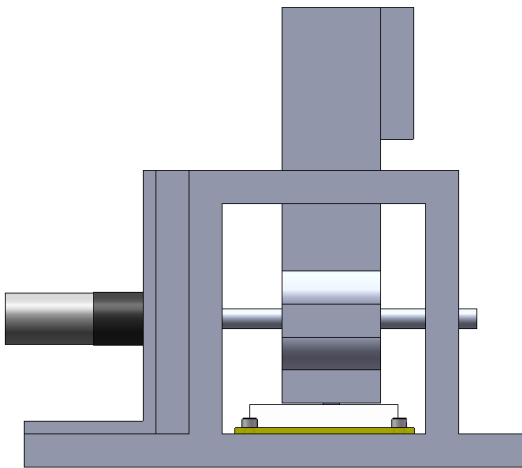
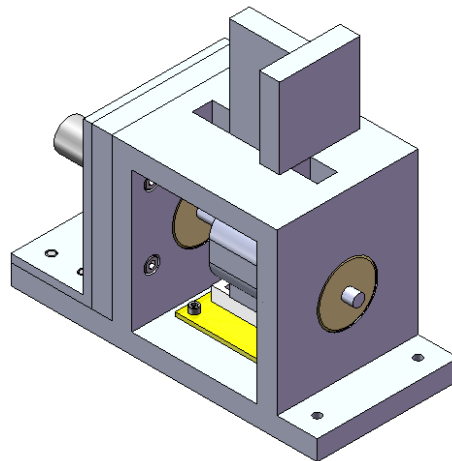


figura 34 - (a) Vista lateral



(b) Vista Isométrica.

A figura 35 mostra o aspecto final do travão, aquele que foi implementado na prática¹.

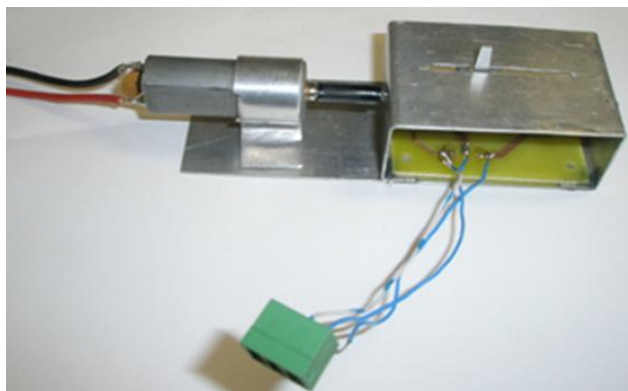


figura 35 – Aspecto do travão na realidade.

Após testes efectuados com carga no setup de testes do travão, concluiu-se que não seria a técnica ideal para accionamento do travão do carro teleoperado.

O travão, embora apresentasse as características necessárias para ser rápido e eficaz na actuação da travagem, pois possui uma velocidade de rotação de 1800 rpm, não apresentava os requisitos de força necessária para conseguir mover o travão real do carro.

Adoptou-se, assim, uma nova técnica baseada no puxar, em detrimento do empurrar (do travão) e, utilizaram-se novos motores. Como o travão não apresentava a força necessária para travar o veículo, pois o motor apresentava um torque de 94g/cm – capaz de permitir uma força máxima de 5Kg com o curso existente no veículo de 25 mm -, alteraram-se os motores, quer o da direcção quer o do travão. Com um torque máximo de 40.8kg/cm, os novos motores cumprem, assim, os requisitos necessários [13].

É ainda de referir que os motores² têm valores de funcionamento na gama compreendida entre 700µs e 2300 µs.

¹ Um agradecimento ao Eng. Fernando Filipe, pela ajuda prestada.

²Referiram-se alguns aspectos dos motores que, numa situação normal, não se teriam em conta, uma vez que o datasheet apenas existe em japonês.

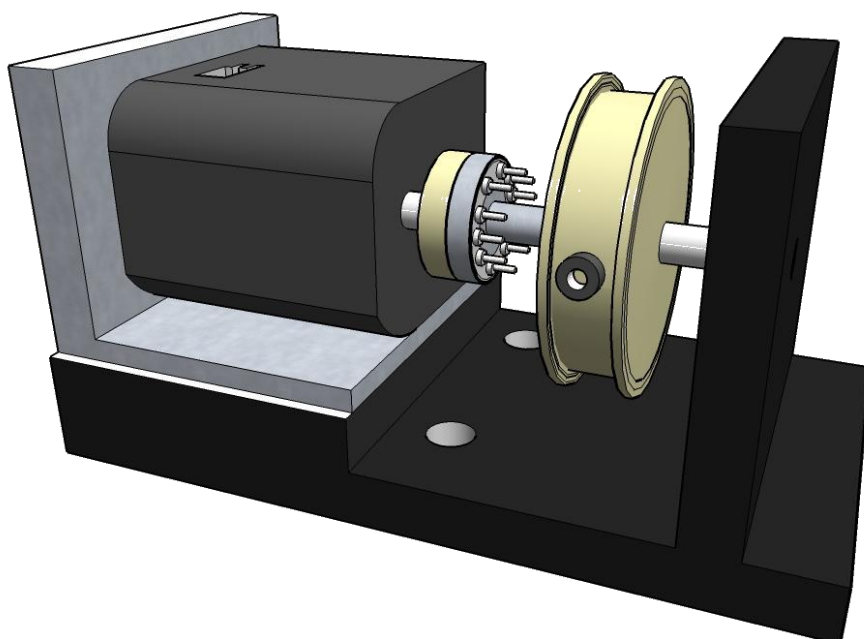


figura 36 - Modelo do travão utilizado.

O travão é apresentado na figura 36 e percorre toda a gama do servo de 0 a 270 graus. O valor 0 equivale ao travão em descanso, correspondendo o valor 270 à força máxima. Preso numa roldana está um cabo com uma mola, que será responsável por puxar o travão existente no veículo.

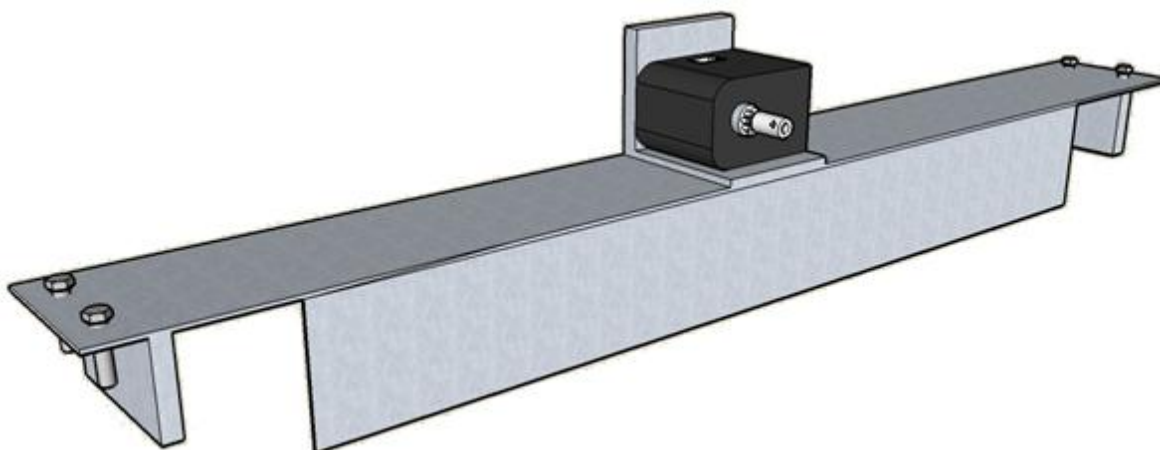


figura 37 - Modelo da direcção utilizado.

Através da figura 37, é possível observar que a direcção apresenta suportes para o encaixe no chassis. O motor ligar-se-á directamente ao veio da direcção do veículo e os ângulos de viragem estarão compreendidos entre -45 e +45 graus, situando-se o ponto 0 nos 135 graus (1500 μ s) do servo.

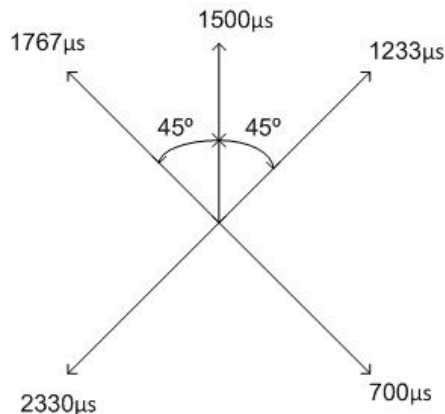


figura 38 - Valores de variação de PWM na direcção.

Os servomotores, por possuírem no seu interior electrónica, não necessitam uma malha de realimentação¹, para o controlo em malha fechada. Será, então, necessário colocar o valor de PWM para o ângulo correspondente, através do microcontrolador.

Os motores funcionam a uma frequência muito mais baixa relativamente à utilizada no ano transacto - 9Khz – frequência essa que era utilizada para modular a alimentação dos antigos motores DC.

Assim, não será possível a utilização directa do módulo incluído na PIC para o PWM, pois a frequência necessária para o funcionamento dos motores será de cerca de 100 Hz, quando o módulo da PIC é utilizado para criar ondas PWM de frequência na ordem dos kHz [12].

Neste sentido, utilizou-se o Timer0 de forma a gerar uma interrupção de 10 em 10 ms e o timer2 para gerar a onda PWM.

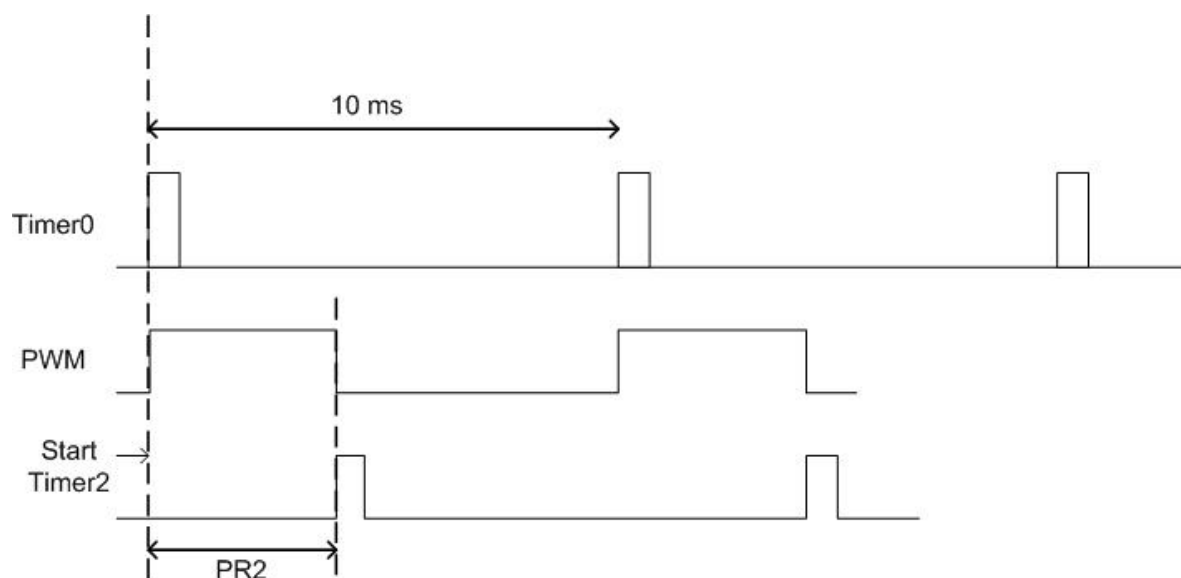


figura 39 - Esquema da criação do PWM.

¹ É sempre possível a existência de sensores, constituindo-se estes como mais um parâmetro de monitorização.

Quando é despoletada uma interrupção de 10 em 10 ms pelo Timer0, o pino que fornece o sinal de controlo PWM é colocado a 1. É calculada, então, a onda PWM necessária e assim, no registo PR2 responsável pelo instante da activação do Timer2, é colocado o valor de quando é que irá ser despoletado uma interrupção pelo Timer2. Quando a interrupção relativa ao Timer2 é activada, é colocado o pino correspondente ao PWM a 0. Este procedimento é efectuado para o controlo do travão, devido a resolução do Timer2.

Para o controlo da direcção a técnica utilizada é análoga, utilizando-se o Timer3 pois possui uma maior resolução.

*Jitter*¹ existente no PWM de controlo:

	Timer3 (direcção)	Timer2 (travão)
Tempo (ns)	210	620

Tabela 2 - *Jitter* no PWM.

Os valores de *Jitter* são muito satisfatórios, não se observando variações nos servomotores, no que diz respeito à mudança de posição – quando estes estão numa posição fixa. Deste modo, garante-se que os motores não sofrem qualquer efeito, provocado por variações no PWM - calculados no microcontrolador.

Alterações no código

Módulo functions

O módulo de controlo não enviava, por CAN, o comando de SCREEN. De forma a corrigir tal situação, alterou-se a função `define_signals` para o comando `screen` ser também enviado.

A tabela de IDs de mensagens CAN a serem enviadas passou a incluir, também, o comando SCREEN.

Módulo cannack

A tabela de envio foi alterada de forma a incluir o comando SCREEN.

¹Variação máxima, relativamente ao tempo ideal.

Conclusões

Embora o sistema de testes do ano anterior tenha ficado funcional, foi necessário efectuar algumas alterações devido à pouca eficácia em termos de força e rapidez que os motores mostravam no controlo da direcção e do travão, quando em controlo remoto, obtendo-se assim uma versão final para colocar no veículo.

Relativamente a alterações ao código, além das que se prendem com o funcionamento dos novos motores, todas as outras foram levadas a cabo, para que o sistema global se mostrasse funcional, sendo agora possível o envio da informação total para o *palmtop/PDA*. A informação, proveniente do volante, passa pelo módulo de controlo tendo este a função de *gateway* volante para o *palmtop/PDA* (através da rede CAN).

A electrónica associada ao Módulo de Controlo/GatewayPalm deixou de possuir a funcionalidade de GatewayPalm, passando “apenas” a módulo de controlo. De uma forma mais pormenorizada, dar-se-á relevo a este assunto num capítulo posterior.

Embora não seja prioritário, pois o sistema encontra-se funcional, seria interessante, numa perspectiva de continuação do projecto, construir-se uma nova placa, com relés em vez de pontes H, de forma a (des)activar os motores (dado que a placa que lá se encontra foi desenhada para os motores DC anteriores). As pontes H permitiam a polarização dos motores DC, além de aumentarem o sinal de 5V, à entrada da ponte, para 12V à saída, disponibilizando mais corrente.

É ainda de referir que os motores utilizados no desenvolvimento deste projecto são alimentados directamente da bateria, permitindo que relés (des)liguem a alimentação dos motores.

Capítulo 5

Software de suporte em palmtop/PDA para arquivo local de informação e disponibilização de dados críticos ao piloto

Neste capítulo analisar-se-á o software existente no *palmtop/PDA*, a rede criada entre os sistemas *palmtop/PDA*, volante e módulo de controlo, bem como o modo de estrutura do código correspondente. Serão, também, abordadas as diferentes topologias de rede que poderiam ser utilizadas no programa final e o motivo pelas quais estas foram inviabilizadas. Descrever-se-á o programa final e a sua implementação, o aspecto gráfico e os requisitos do programa.

Introdução teórica

Tecnologias de Suporte à comunicação

Bluetooth

Bluetooth é uma especificação industrial para áreas de redes pessoais sem fios (*Wireless personal area networks* - PANs). O bluetooth permite conectar e trocar informações entre dispositivos, nomeadamente telemóveis, portáteis, computadores, impressoras, câmaras digitais, entre outros.

Os dispositivos bluetooth operam na faixa ISM (Industrial, Scientific, Medical) centrada em 2,45 GHz. Nos Estados Unidos, a faixa ISM varia de 2400 a 2483,5 MHz. Na maioria dos países da Europa, a mesma banda também está disponível. No Japão a faixa varia entre 2400 a 2500 MHz.

Os dispositivos são classificados de acordo com a potência e alcance, em três níveis como mostra a tabela seguinte:

Classe	Potência máxima (mW/dBm)	Alcance (valor aproximado)
Classe 1	100 mW (20 dBm)	~ 100 metros
Classe 2	2.5 mW (4 dBm)	~ 10 metros
Classe 3	1 mW (0 dBm)	~ 1 metro

Tabela 3 - Especificações do bluetooth.

A banda ISM encontra-se dividida em 79 portadoras, espaçadas 1 MHz, podendo cada dispositivo transmitir em 79 frequências diferentes. De forma a minimizar as interferências, o dispositivo "master", após sincronizado, pode mudar as frequências de

transmissão dos seus "slaves" até 1600 vezes por segundo. Em relação à sua velocidade pode chegar a 721 Kbps e possui três canais de voz.

Cada dispositivo é dotado de um número único - 48 bits - que serve de identificação, no formato 00:00:00:00:00:00, denominado "Endereço de Bluetooth" (*Bluetooth Address*) e são únicos e exclusivos para cada dispositivo fabricado, assim como o endereço MAC da placa de rede.

Os dispositivos bluetooth comunicam-se entre si e formam uma rede designada **piconet**, na qual podem existir até oito dispositivos interligados, pois *piconets* têm um espaço de endereçamento de 3 bit, o que limita o tamanho máximo da *piconet* a 8 dispositivos ($2^3 = 8$), sendo um deles o mestre – *master* - e os outros dispositivos escravos – *slave*.

A junção de 2 ou mais *piconets* constituem uma **scatternet**. Esta permite o suporte de comunicação de mais do que 8 dispositivos. As *scatternets* podem ser formadas quando um elemento de uma *piconet* (*master* ou *slave*) participa noutra *piconet* em separado. A participação do dispositivo em ambas as *piconets* permite a troca de informação entre elementos das *piconets*. Usando esta filosofia é possível juntar várias *piconets* numa complexa *scatternet*, expandindo assim o tamanho físico da rede bluetooth para além dos seus limites impostos.

Sistema anterior – PalmOS

O programa de monitorização anteriormente desenvolvido para o ICARO, foi desenvolvido para palmOS, considerando-se já a hipótese de se construir uma *piconet* com vários dispositivos¹. O dispositivo armazenava, então, a informação para *logging* e futuro visionamento no monitor do ICARO, apresentando a informação mais relevante ao piloto, nomeadamente em termos de velocidade instantânea, distancia percorrida, número da volta, estado da ligação bluetooth, o estado da bateria, RPM do motor, o tempo de prova vs tempo ideal de prova – a 30km/h.



figura 40 - Imagem do programa monitor, em PalmOS.

¹API (Application Programming Interface). Esta estrutura permitia fazer directamente a *piconet*.

É de referir que, a nível gráfico, a constituição deste programa mostrou-se uma ajuda fundamental, na medida em que possibilitou definir mais claramente a ideia do formato em que se queria o programa, para visualização ao piloto. No que diz respeito à construção do código, e consequente reutilização deste, não era possível devido, essencialmente, à criação da rede, uma vez que se utilizaria o sistema operativo Windows Mobile 5.0.

Constituição das tramas existentes nas redes sem fios

As tramas têm o seguinte aspecto:

00000000	Header 0
1NNNNNNN	Header 1 (Num. De bytes do restante da trama < 72)
BBBSSSSS	BBB = Board SSSS = Signal ID
DDDDDDDD	Data
.....	
BBBSSSSS	
DDDDDDDD	Em caso de dados multi-byte
DDDDDDDD	Adoptar formato big-endian
CCCCCCCC	CRC 0
CCCCCCCC	CRC1

IDs das mensagens circulantes nas ligações sem fios:

Placa	BoardID
GatewayPalm	0
GatewayBoxes	1
GPS	2
Roda	3
Motor (monitorização)	4
Motor (controlo)	5
Volante	6

Tabela 4 - IDs das placas.

Para obter mais informações acerca das tramas, consultar o anexo B.

Arquitectura do sistema

O *palmtop/PDA* acer n300, por possuir o Windows mobile 5.0 como sistema operativo, não permite, à partida, efectuar uma *piconet*. Para tal, utilizou-se um sdk¹ (*software development kit*) para o desenvolvimento e criação de aplicações bluetooth, com base em .net.

No ano transacto realizaram-se apenas alguns testes, com o intuito de verificar se o volante se ligava ao pda e se havia perda de tramas, não sendo possível obter a rede com o *palmtop/PDA* – master -, volante e módulo de controlo – slaves -, faltando ainda construir a rede entre os 3 dispositivos. A criação desta rede permite que se efectuem trocas de informação entre todos os dispositivos.

A rede tem de ser capaz de permitir a troca de informação entre o volante, o *palmtop/PDA* e o módulo de controlo, sendo o *palmtop/PDA*, a *gateway* entre o volante e o módulo de controlo.

Numa perspectiva teórica, existem 3 hipóteses para a criação da rede:

- *Piconet* com *palmtop/PDA* como *master*; volante e módulo de controlo como *slaves*;
- *Piconet* com um dos módulos bluetooth existentes no volante ou módulo de controlo como *master*; *palmtop/PDA* e o outro módulo bluetooth como *slaves*;
- *Scatternet* com *palmtop/PDA* como *slave* (sendo o volante o *master*) e *palmtop/PDA* *master* (módulo de controlo como *slave*) – ver figura 41.

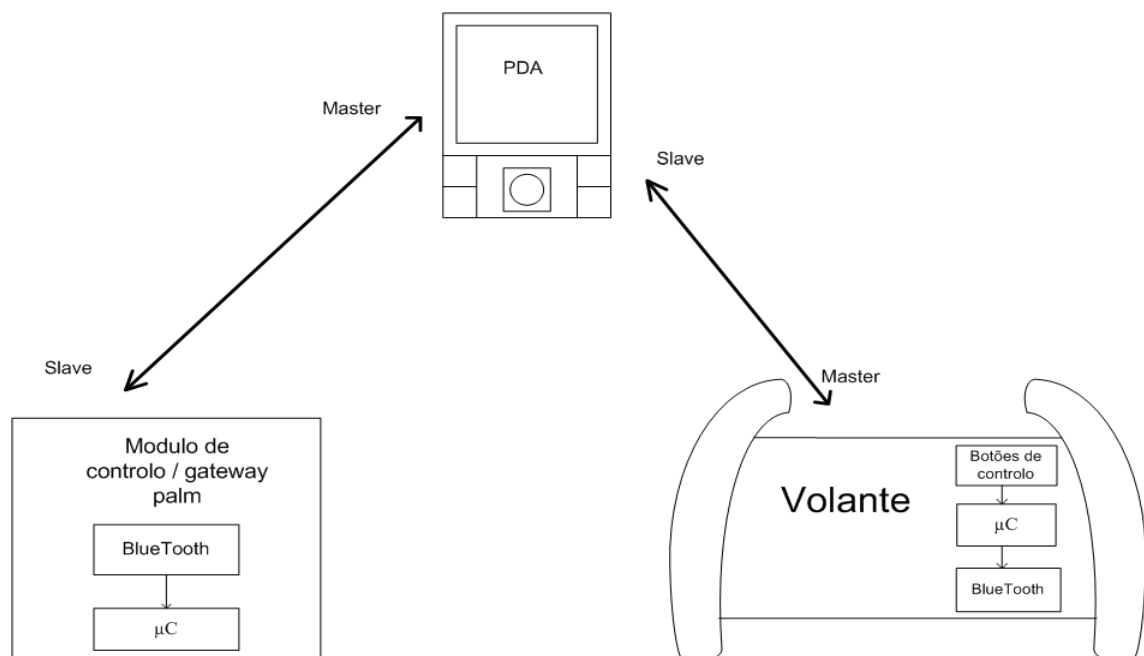


figura 41 - Ligação sem fios bluetooth do sistema – *scatternet*.

¹ Frason Bluetools SDK.

De seguida analisar-se-á cada hipótese em particular, bem como vantagens e desvantagens associadas a cada uma e a sua implementação prática¹.

Piconet – palmtop/PDA como master, volante e módulo de controlo slaves

Uma vez que no ano transacto já se tinha tentado criar a rede, embora não tenha sido possível, o programa utilizado funcionava noutros *palmtop/PDA* contudo, não funcionava no acer n300, mostrando-se tal facto como indicador de que o *palmtop/PDA* já possuía algumas limitações ao nível do *hardware*.

Este ano foi já possível criar a rede com o *palmtop/PDA* como *master*, transmitindo a informação entre o volante, o módulo de controlo e o *palmtop/PDA*. A grande vantagem de ter o *palmtop/PDA* como *master* na *piconet*, era a possibilidades de, através das bibliotecas Frason [14], voltar a conectar e implementar uma máquina de estados em caso de perda de ligação. Neste caso o *palmtop/PDA* seria totalmente responsável pela tentativa de voltar a ligar a qualquer dos módulos.

De forma a facilitar as ligações, a partir de uma base fiável de funcionamento, o software foi adaptado dos exemplos do Frason Bluetools SDK. Assim, tal como se vê através da imagem², é possível descobrir os dispositivos, bem como os seus serviços (neste caso específico, a transmissão de informação é o “serial port service”), em paralelo com o envio/recepção de informação nas duas ligações.

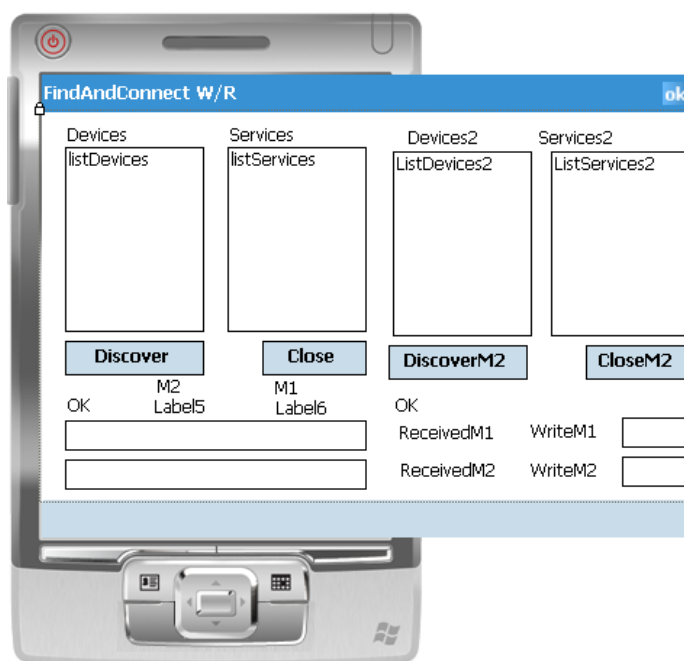


figura 42 - Programa de testes.

¹ O código será explicado somente na versão final adoptada para o software pois, além de se mostrar muito repetitivo, não permitia centrar a atenção no que é realmente importante - a versão adoptada e final. O respectivo código encontra-se no CD anexo à tese, para futura análise e possível utilização.

² Como seria de esperar, a tela do *palmtop/PDA* não é tão grande como a tela criada para visionamento, sendo necessário usar uma barra de deslocamento horizontal.

O diagrama seguinte representa a criação da rede:

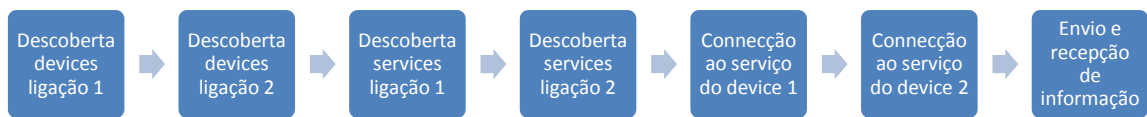


figura 43 - Criação da rede.

Porém, a criação da rede desta maneira apresenta alguns problemas. Entre os quais destacam-se:

- a não existência de, em caso de perda de ligação, possibilidade de restauro, uma vez que o *palmtop/PDA*, emite erro de “discover device timeout”, quando existem 2 ligações¹;
- a descoberta dos dispositivos tem que ser feita uma a seguir a outra, não existindo a possibilidade de ligar a um dispositivo e depois ao outro, pois o *palmtop/PDA* devolve sempre erros na descoberta do 2º dispositivo, quando o 1º já se encontra ligado;
- a pouca fiabilidade, o que não permite a criação de uma máquina de estados para ligar novamente em caso de perda de conexão.

Apesar de tudo isso, em testes efectuados com o envio de 10000 tramas de 8 bytes (o pc ligado como substituto do módulo de controlo), para o *palmtop/PDA*, simultaneamente com o volante a enviar informação para este, não se perdeu qualquer trama.

Porém, a grande vantagem de criar uma *piconet* com o *palmtop/PDA* como *master*, sendo este o responsável pela gestão da rede, não se verificou, uma vez que não é fiável, sendo portanto colocada de lado esta hipótese para a criação da rede.

Piconet – Módulo de controlo master, *palmtop/PDA* e volante slave

Os testes realizados deixaram a descoberto que não era necessário utilizar o *palmtop/PDA* como *gateway*, pois o volante comunicava directamente com o MC, sendo *palmtop/PDA* somente para visionamento e *logging* de informação.

Em relação aos módulos bluetooth existentes no volante e MC só permitem ligações ponto-a-ponto não sendo por isso possível fazer a *piconet*.

¹O programa somente com uma ligação, permite a desconexão e reconexão sem qualquer tipo de problemas, o que não se verifica com 2 ligações, quando o *palmtop/PDA* está como *master*.

Scatternet – Volante Master – PDA slave e PDA Master-MC slave

A rede foi construída com duas ligações, sendo que uma delas era feita pelo *palmtop/PDA* (para o MC) e a outra criando um serviço serial port com o Franson Bluetools SDK, ao qual o volante se ligava.

Diversos testes efectuados revelaram que não se perdia qualquer trama nesta configuração. Porém, o processamento das tramas revelou-se muito lento, existindo um atraso de cerca de meio segundo quando as duas ligações estavam em simultâneo a enviar informação.

Mais tarde, ao invés de se criar um serviço pelo Bluetools, acedeu-se simplesmente à porta serial do *palmtop/PDA* ao qual o volante se ligava. Diferentes testes mostraram que, embora a informação da serial port fosse rapidamente processada, os dados provenientes do buffer do Bluetools não eram processados a tempo, existindo uma grande perda de tramas. Em 10000 tramas de 8 bytes enviadas sendo o *palmtop/PDA master* apenas 3648 tramas foram recebidas.

Outro aspecto negativo a apontar a esta hipótese é a inexistência da possibilidade de saber quando ocorria a perda de ligação, algo que com a criação de serviço era possível de saber, no que diz respeito à recepção de dados. Esta situação mostrava-se impraticável pois o *palmtop/PDA* serviria de gateway do volante para o MC, havendo assim perda de controlo do veículo.

O quadro apresentado de seguida, mostra-se como resumo das principais características das diversas topologias de rede.

	Fiável (possibilidade de restauro da ligação)	Perda de tramas	Rapidez em processamento das tramas para gateway
<i>Piconet</i> (PDA – Master, Volante e MC slaves)	Não	Não	Não
<i>Piconet</i> (MC (ou Volante) – Master, Volante (ou MC) e PDA slaves)	Não é possível criar	Não é possível criar	Não é possível criar
<i>Scatternet</i> – Volante Master – PDA slave. PDA Master-MC slave (via Frason Bluetools)	Sim	Não	Não
<i>Scatternet</i> – Volante Master – PDA slave. PDA Master-MC slave (via porta COM)	Sim	Sim	Não

Tabela 5 - Principais características das ligações efectuadas.

O maior problema das ligações efectuadas é conseguir que os comandos de controlo circulem entre o volante e módulo de controlo, servindo o *palmtop/PDA* como gateway, não havendo perda de tramas. Tal facto não é verificável com a utilização deste *palmtop/PDA*, uma vez que este processa cerca de 1 trama de 250 em 250 ms quando as

2 ligações estão activadas, o que implica bastantes problemas a qualquer destas topologias de rede.

Mudança da topologia da rede

Com o objectivo de evitar que o *palmtop/PDA* servisse de gateway e torná-lo apenas um sistema terminal de *logging* e visionamento, alterou-se a topologia da rede. Esta alteração permite minimizar o atraso entre o volante e o módulo de controlo, bem como a necessidade de processamento do *palmtop/PDA*.

A rede utilizada foi a descrita através da figura 44:

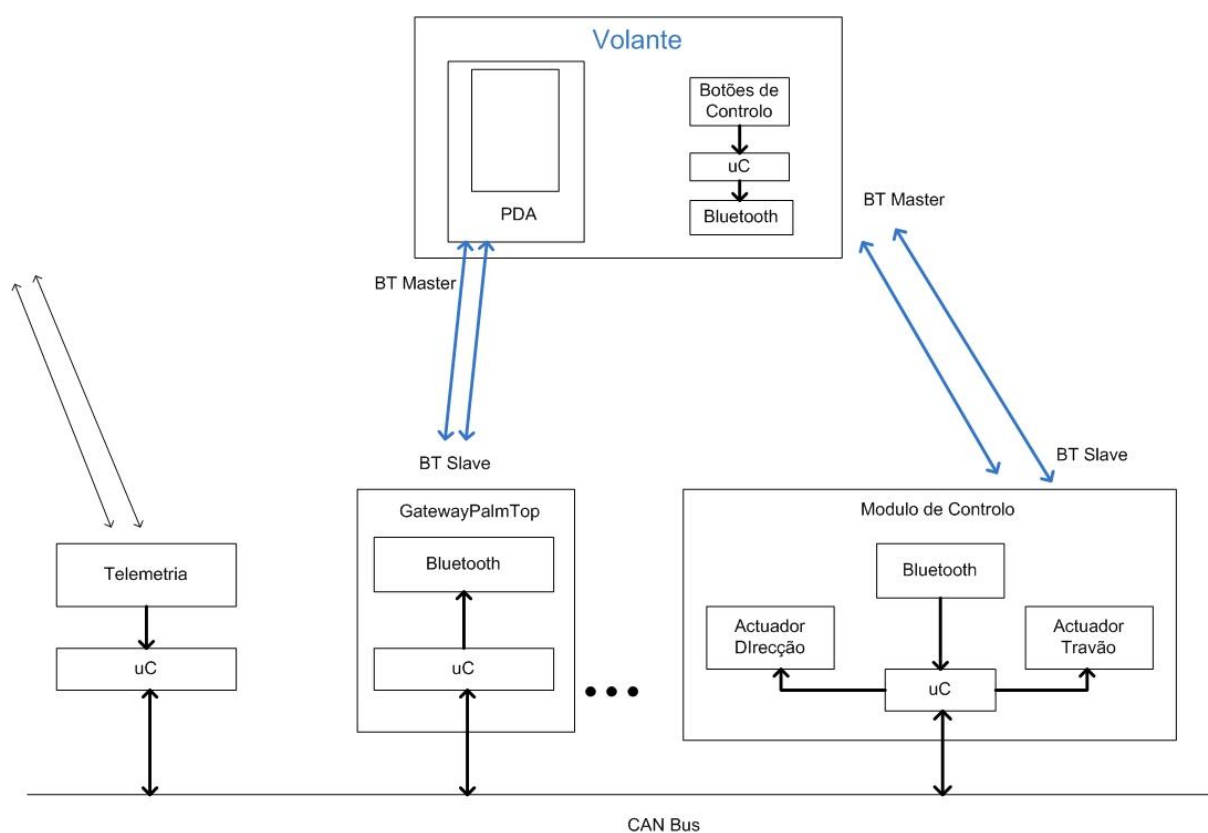


figura 44 - Arquitectura global ICARO com a nova topologia.

Como se pode verificar na figura 44, o módulo de controlo/GatewayPalm deixa de ter as 2 funções, apresentando-se em separado, existindo agora uma nova placa com a função GatewayPalm (será referido no capítulo seguinte). Esta placa estará na rede CAN e tornará possível a existência do *palmtop/PDA* como ponto terminal servindo a placa como electrónica de suporte ao *palmtop/PDA* para recebimento de dados.

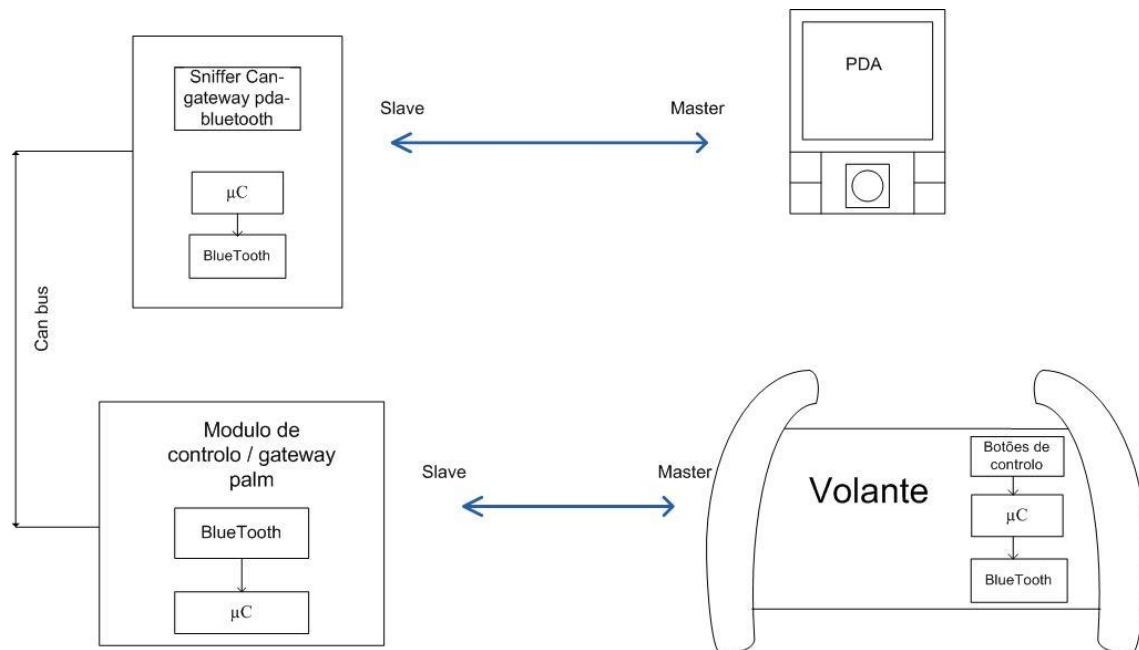


figura 45 - Nova topologia da rede – mais detalhada.

A placa terá a função de sniffer da rede CAN enviando, posteriormente, por bluetooth a informação em tramas para o *palmtop/PDA*, permitindo assim “ancorar” o *palmtop/PDA* no sistema, sem interagir directamente com este, tornando-o completamente independente do *palmtop/PDA*.

Programa Final do palmtop/PDA

Aspectos gerais

O *palmtop/PDA* será capaz de:

- iniciar a ligação bluetooth com um dispositivo e receber informação;
- possuir capacidade para, em caso de perda de ligação, voltar a conectar ao dispositivo (máquina de estados);
- validar a informação (tramas) vinda do bluetooth (máquina de estados);
- processar os dados (efectuar o log deles, visualizar a informação, fazer alguns cálculos com eles, entre outros) através de uma máquina de estados;
- criar um ficheiro de *log* dos dados recebidos para futura visualização no monitor do ICARO;
- criar um ficheiro de *log* de erros para registo de possíveis falhas ocorridas (a título de exemplo, o aparecimento de caixas de erro ao condutor não permitiria ver o que realmente importa – a informação);
- configurar a ligação bluetooth e escolher dispositivo a ligar, sem ser necessário alterar código (através da criação de um ficheiro de configuração de bluetooth);

- desenhar um gráfico de perfil de altitude, de forma a que o condutor saiba o local onde se encontra.

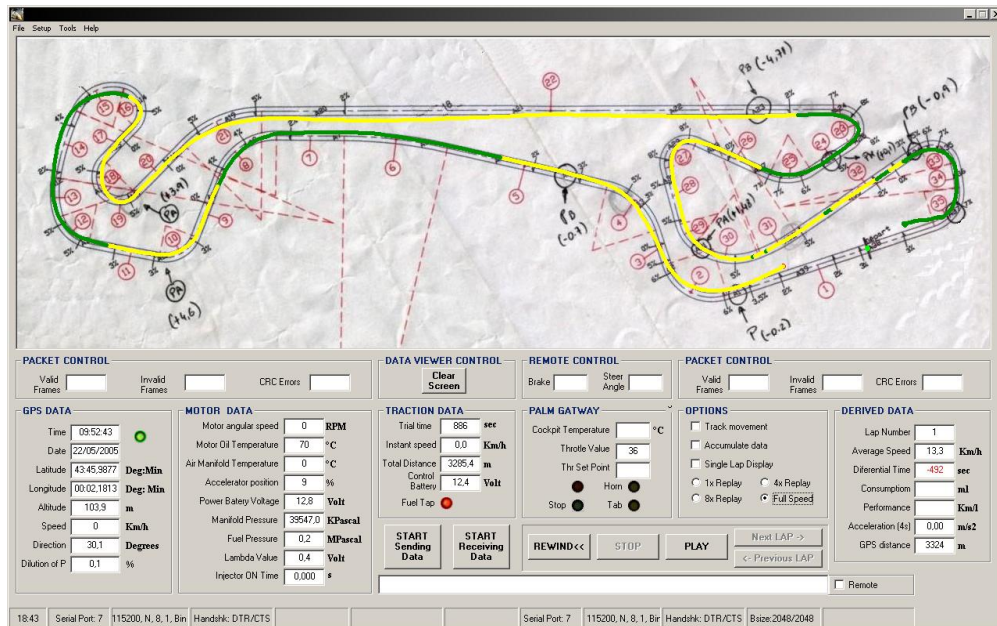


figura 46 - Monitor ICARO.

Ligação Bluetooth e obtenção dos valores para posterior processamento

A ligação bluetooth será realizada através das bibliotecas de desenvolvimento de aplicações Frason Bluetools. Estas bibliotecas permitem a ligação a um serviço de um determinado dispositivo, bem como a descoberta de dispositivos e serviços. A abertura e manutenção de uma ligação e a disponibilização de funções de leitura e de escrita para um *buffer* de valores provenientes da ligação, são também aplicações das referidas bibliotecas.

Depois da informação se encontrar disponível no *buffer* do bluetooth, esta é copiada para um *buffer* circular de modo a, por um lado não se perder informação devido a possíveis rajadas de bytes provenientes do dispositivo, por outro, caso se percam alguns dados, quando o ponteiro de escrita do *buffer* circular igualar o ponteiro de escrita, perder-se-ão os dados mais antigos.



figura 47 - Aspecto da recepção de dados.

O *buffer* circular tem um tamanho de 8kbytes e possui um ponteiro de escrita e de leitura que apontam para o 1º elemento livre para escrita e leitura, respectivamente.

Processamento das tramas

As tramas existentes no *buffer* circular têm que ser processadas. Para tal utilizou-se um *timer* que é activado de 100 em 100ms, que verifica se existem *bytes* no *buffer* circular para serem processados. Em caso afirmativo, a máquina de estados seguinte é activada (função *processdata*), até que todos os *bytes* do *buffer* circular tenham sido processados, isto é, ponteiro de escrita = ponteiro de leitura.

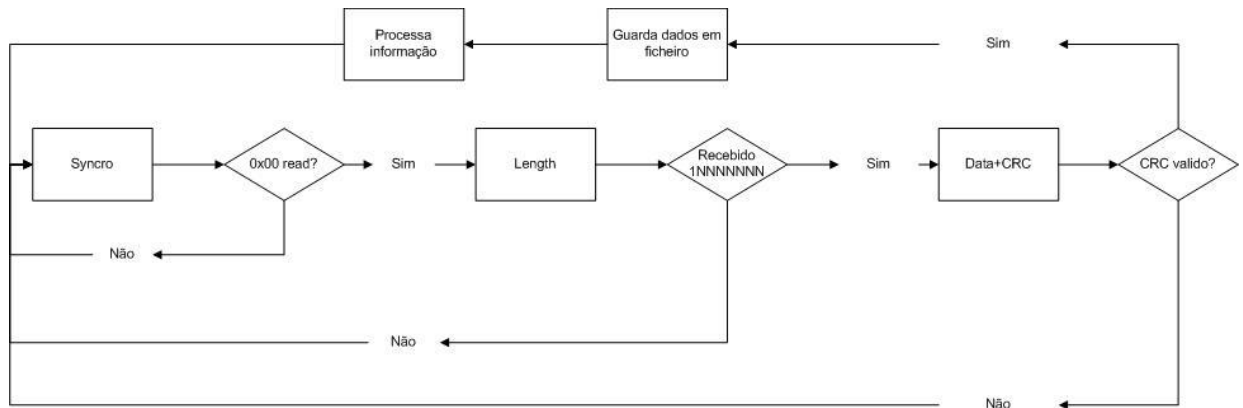


figura 48 - Máquina de estados - process_data.

Existem 3 estados em que é possível encontrar a função:

- Syncro (no programa realizado denominado por Header_0)
- Length (no programa realizado denominado por Header_1)
- Data+CRC (no programa realizado denominado por Data)

O estado Syncro é responsável por “capturar” o início de cada trama, passando para o estado seguinte, caso tenha sido recebido 0x00. Em caso negativo, continua neste estado até se capturar o que poderá ser o início de uma trama.

O estado *length* é responsável por dar conta de quantos *bytes* de dados (incluindo CRC) virão a seguir e até ao final da trama. É de salientar ainda que cada trama terá, no máximo, 72 *bytes* e que as tramas poderão ter vários sinais de forma a diminuir o *overhead*. Assim, no estado *length*, existe um *byte* com a estrutura 1NNNNNNN, sendo N o número de *bytes* restantes da trama (< 72 bytes). Em caso negativo, retoma-se o estado *syncro*.

O estado final Data+CRC é responsável por armazenar a informação em 2 buffers:

Buffer possuindo a trama completa, com o *header* de *syncro* e o *header* de *length*, juntamente com os 2 bytes de CRC e, claro, os dados. Este *buffer* tem como finalidade o armazenamento de tramas válidas para futuro processamento dos dados noutra máquina de estados. Essa máquina de estado será referida mais a frente e tem a finalidade de processar *data*, visualizar, fazer cálculos, entre outros.

Buffer para guardar dados, num ficheiro para futuro visionamento no monitor do ICARO¹, possuindo apenas os dados e respectivos identificadores dos sinais. Não se utilizou o mesmo buffer, pois o 1º referido é utilizado para processamento de dados, enquanto que o 2º têm uma capacidade maior e é um buffer temporário onde os dados são colocados até que se tenham mais de 1024 bytes. Nesse momento, os *bytes* são escritos no ficheiro de dados criado no SD-CARD do *palmtop/PDA*.

Esta técnica foi utilizada de forma a evitar a escrita constante no cartão de memória, guardando apenas quando já se tem bastante informação².

Além disso, o estado Data+CRC possui a função de verificar a validade das tramas através do CRC sendo que, em caso afirmativo, a informação é guardada nos 2 buffers acima referidos.

Processamento dos dados

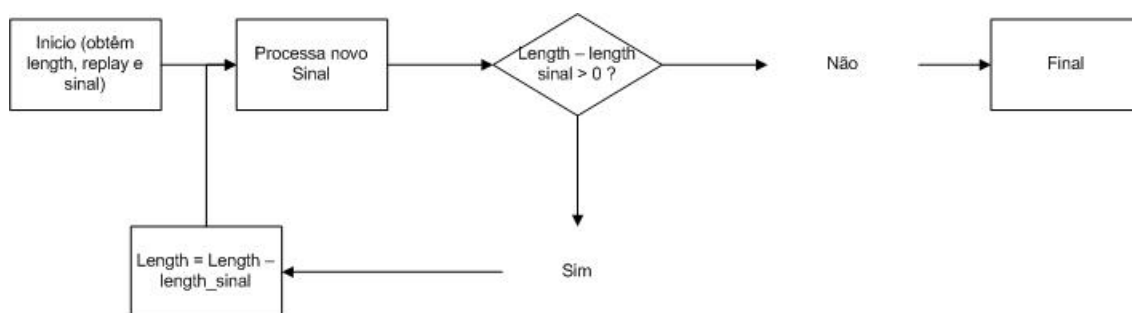


figura 49 - Máquina de estados - *extract_data*

No estado Data+ CRC referido anteriormente, em caso de trama válida, é chamada a função que é responsável pelo tratamento da informação (função *extract_data*). Essa função parte do 1º *byte* existente no buffer - que é um identificador – e, sabendo quantos *bytes* de dados relativos a esse sinal vêm a seguir, faz o que tiver de ser feito de acordo com esse sinal (por exemplo, imprimir no ecrã, fazer cálculos, etc.). Este ciclo continuará a ser executado até que todos os *bytes* da trama válida tenham sido processados.

Aspectos do programa principal

Form1.vb é o programa principal, onde são efectuadas as operações fundamentais com base nos módulos existentes, onde é feito o *load do* programa principal e onde se termina o referido programa. Todos os restantes módulos são auxiliares, de forma a permitir que o Form1 esteja mais perceptível para leitura e compreensão. O programa apresenta 6 módulos [14], [15], [16]:

¹O programa de *palmtop/PDA* possui um “mini-monitor” que permite o visionamento do ficheiro de dados - a prova.

²Surgindo como excepção o final do programa que, independentemente de se ter 1024 *bytes*, guarda a informação que ainda exista no buffer.

- **Blue_funcs.vb** – contempla as funções para o correcto funcionamento do bluetooth, desde a função de escrita, leitura de e para um dispositivo bluetooth e as funções de connect_device() e de disconnect_device() que o nome já é explicatório¹;
- **Buffers_funcs.vb** – possui as funções de leitura e escrita num buffer e 2 funções de processamento de tramas, uma para dados vindos de um dispositivo bluetooth, outra para os dados oriundos de um ficheiro;
- **CRC.vb** – contém a função para análise do CRC vindo das tramas;
- **Filters.vb** – contempla as funções para os valores que necessitam de cálculos (distância, velocidade, etc.) e as funções para a passagem de valores numéricos para a correspondente imagem para visualização. As funções relativas a dados que não são visualizados, embora funcionais, não são executadas durante o programa para não haver processamento desnecessário. Este módulo só é chamado pelo módulo signals.vb;
- **Signals.vb** – é o módulo responsável pelo tratamento dos dados. Existem 37 sinais possíveis para tratamento, alguns dos quais é necessário apenas visualizar, outros fazer cálculos e outros somente incrementar o buffer para os dados seguintes;
- **Include.vb** – possui, como o nome indica, as variáveis globais ao sistema, bem como os parâmetros de definições gerais (tamanho de buffers, ID das placas, etc.).

¹O reconnect_device não foi efectuado dentro deste módulo mas sim directamente no programa pois trata-se de um disconnect_device – connect_device, que é efectuado quando o buffer de bluetooth retorna um erro de ErrorCode.ConnectionFailed.

O programa principal do monitor HammerShark terá o seguinte aspecto:

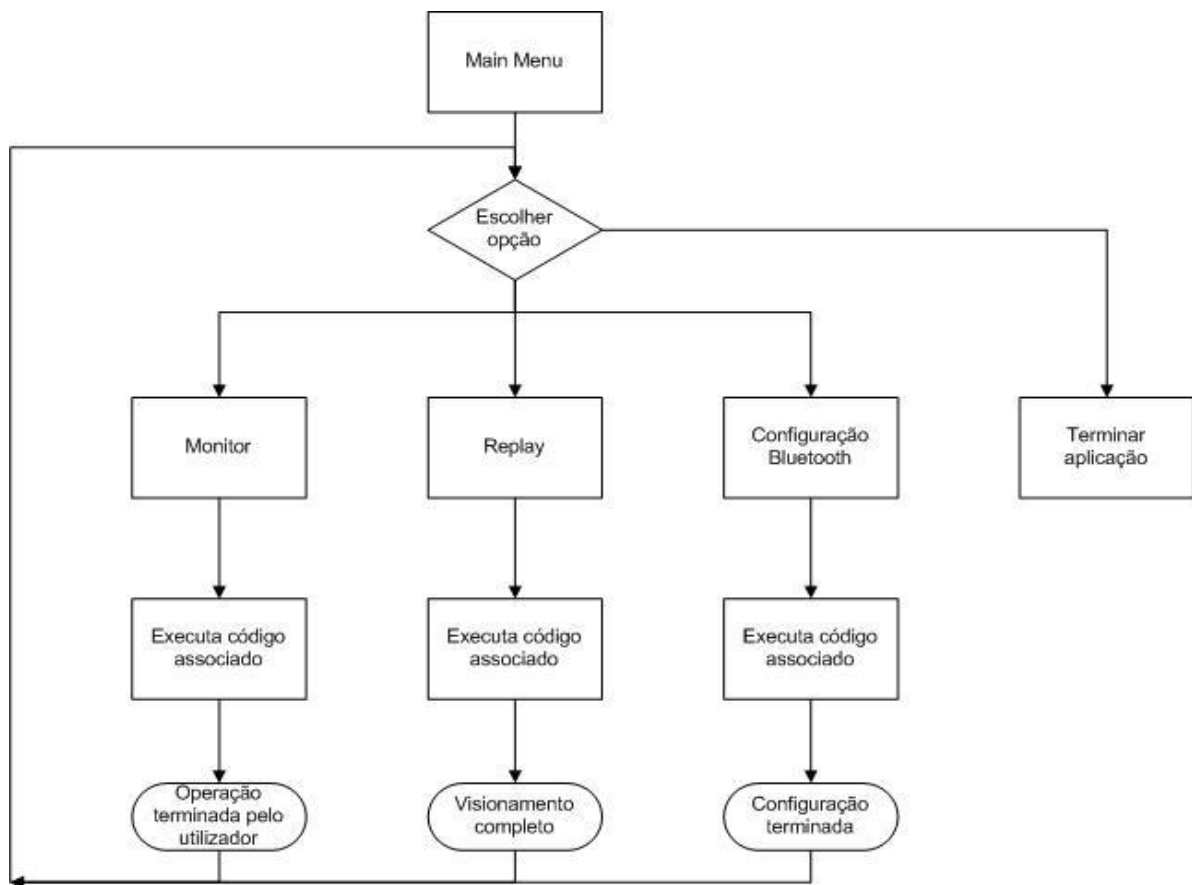


figura 50 - Máquina de estados - programa principal.

Tal como se verifica através da figura acima, o menu principal possibilita apenas a escolha da operação que o utilizador deseja fazer.

Já o menu monitor, por sua vez, é constituído por 2 ecrãs que disponibilizam informação útil, em tempo-real, ao utilizador. Num dos ecrãs, é fornecida informação acerca da velocidade, da distância percorrida, número da volta, estados das baterias, etc. O outro menu possui um perfil de altitudes que permite ao piloto saber se está a subir ou a descer, quais são as subidas e as descidas mais acentuadas, entre outros. Esses 2 painéis são mudados através do carregar de um botão existente no volante do veículo. O menu monitor é também responsável pela ligação bluetooth com um dispositivo já pré-determinado (existe um ficheiro de configuração no cartão de memória com os parâmetros do dispositivo ao qual se pretende ligar) para, em caso de falha, tentar novamente reconectar ao dispositivo e também pela salvaguarda da informação num ficheiro.

O menu replay permite abrir um ficheiro de dados à escolha do utilizador e rever a prova, tal como é mostrada no menu monitor ao piloto. Além disso, permite fazer *pause/play* aos dados da prova em questão.

No que diz respeito ao menu configuração, este permite fazer a descoberta de dispositivos na área de alcance do bluetooth, os seus serviços e ligar aos dispositivos. Caso o utilizador pretenda, é possível definir o dispositivo seleccionado como dispositivo por defeito a ser utilizado na próxima vez que o menu monitor for iniciado.

Ficheiros criados pelo monitor e de configuração

Diversos ficheiros são manipulados e utilizados pelo programa, entre os quais:

Ficheiros de configuração:

- **Bluetooth_conf.txt** – possui todos os dados do dispositivo e serviço ao qual o programa se liga automaticamente. É possível de ser alterado no menu configuração, através da descoberta de dispositivos e seus serviços.
- **Track_info.txt** - possui os dados relativos á pista. Esses dados são: distância total, número de voltas, tempo de prova e comprimento de cada volta.

Ficheiros de dados:

- **icaro_data_tempo.txt** – contêm toda a informação recebida pelo *link* bluetooth. Apenas é criado quando o utilizador selecciona o menu Monitor. Data e o tempo é a data actual dada em ano, mês e dia tal como o tempo é dado em hora, minutos, segundos de forma a não existir repetição de nomes.
- **icaro-error_log_data_tempo.txt** – ficheiro de *logging* de estado do programa e erros que possam eventualmente acontecer.

Aspecto Gráfico

Apresentam-se, de seguida, os menus e sua aparência gráfica, bem como a forma de tratamento da informação para mostrar ao piloto, uma vez que a informação terá de ser espelhada para visualização.

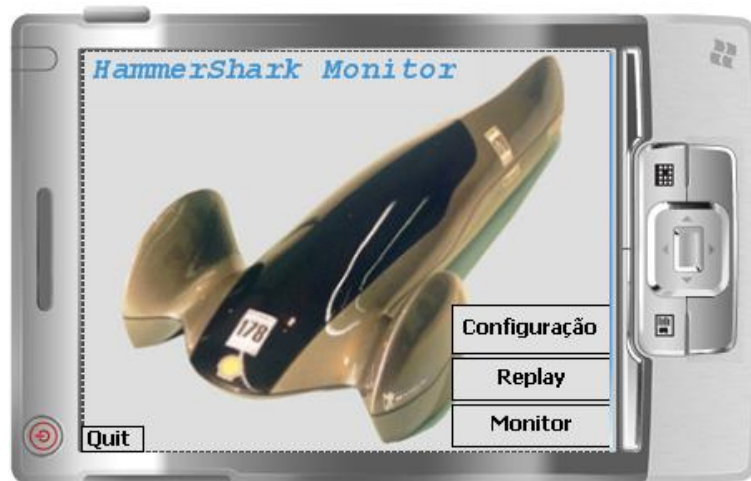


figura 51 - Menu Principal.

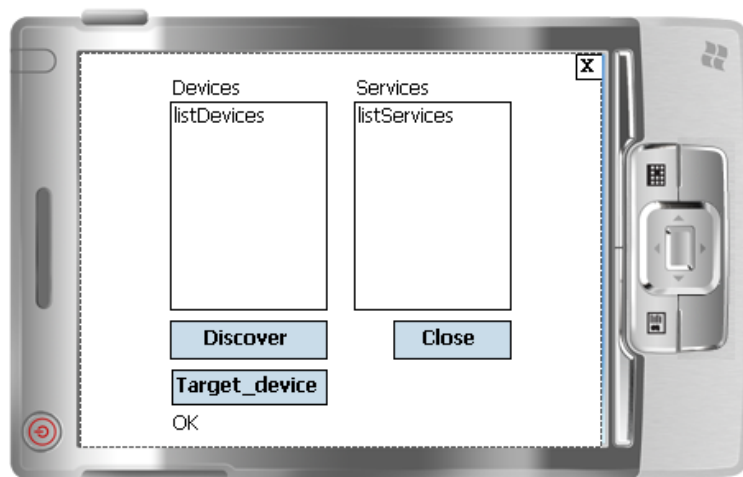


figura 52 - Menu Configuração Bluetooth.

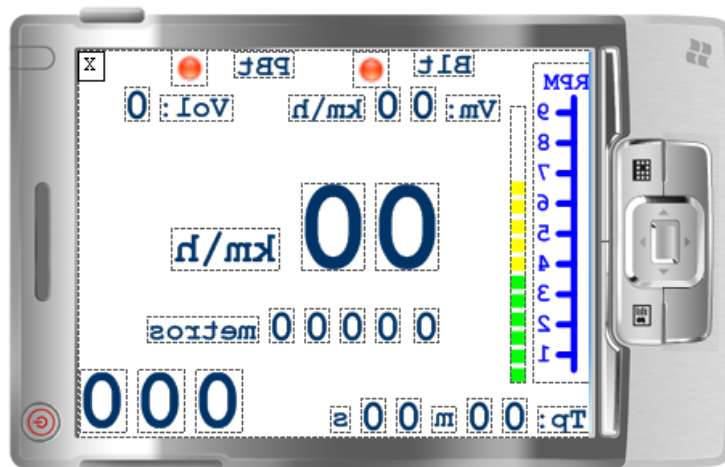


figura 53 - Menu Monitor/replay.

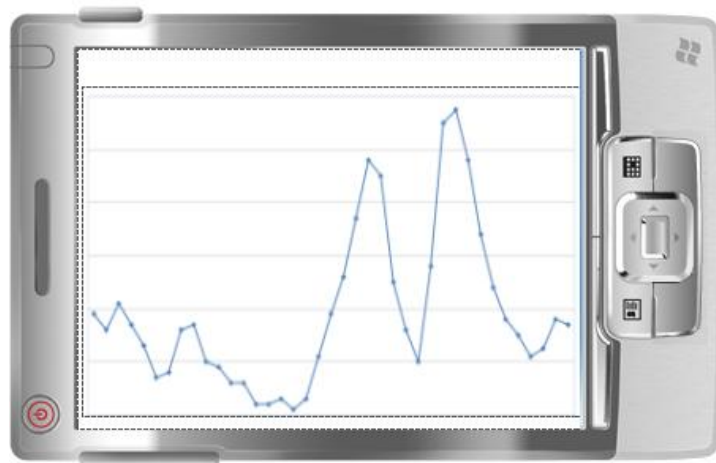


figura 54 - Menu Altitudes (acedido somente do menu monitor/replay).

O menu Monitor (menu número 3) é iniciado quando o utilizador pressiona o botão *replay* ou monitor, ainda que com algumas diferenças. No menu Monitor relacionado com o *replay*, é pedido ao utilizador que seleccione um ficheiro para visualização existente no SD-CARD do *palmtop/PDA*.

Já no menu Monitor relacionado com a monitorização ao piloto, é efectuada a ligação bluetooth, através de um ficheiro de configuração existente no SD-CARD¹ (passível de ser alterado pelo menu de configuração bluetooth). Deste modo, é feita a visualização da informação e *logging* dos dados recebidos no SD-CARD do dispositivo. Além disso, é criado um ficheiro de erros e *status* acerca de eventuais erros, conexão, entre outros².

Como se pode ver, o menu Monitor possui a informação mais relevante: velocidade, distancia percorrida, numero de volta, tempo de volta etc...

Além do já referido, foi colocado um menu de altitudes da pista para o piloto ter conhecimento acerca da topografia do terreno, nomeadamente se deve desligar o motor, se está a descer ou a subir, possíveis subidas e descidas acentuadas, entre outras informações de interesse.

Uma vez que a informação vai passar por um espelho antes de ser visionada pelo piloto, toda a informação destes 2 menus encontra-se espelhada.

¹Esse ficheiro contém a informação acerca do endereço do dispositivo bluetooth, o nome do dispositivo, o nome do serviço a que se deseja ligar, bem como o número do serviço respectivo.

²Para o piloto, a informação relativa aos erros não é muito relevante, pelo que estas mensagens são colocadas num ficheiros para posterior processamento nas boxes.



figura 55 - *Palmtop/PDA* em cima - espelho para visualizar a informação[1].

É de referir que o *palmtop/PDA* encontra-se colocado em cima do volante, existindo um espelho que permitirá ao piloto ver essa informação.

Um outro aspecto que se verificou é que o *palmtop/PDA* suporta o tempo da prova, não falhando a bateria. Tal facto é passível de ser comprovado pois, ao enviarem-se tramas de 8 bytes para o *palmtop/PDA*, durante 1 hora, verifica-se que este apenas “gastou” cerca de 26% da bateria.

Opções tomadas a nível visual [17] [18]

Relativamente ao aspecto gráfico, mais propriamente ao menu Monitor e Gráfico, existiam 2 hipóteses para a sua criação em vb.net:

- um conjunto de imagens agrupadas numa *Image_List* que permite mostrar numa *picturebox*, e repetir a visualização quantas vezes se pretender e em quantas *pictureboxs* se quiser;
- sobrepor todas as imagens que se quiserem (por exemplo colocar as imagens de 0 a 9 invertidas e sobrepostas) e tornar visível a que se quer e invisível as restantes. Deste modo garante-se que só a que queremos é mostrada ao utilizador. Cada grupo de imagens sobrepostas poderão ser agrupadas num *array* de *picturebox*, de forma a tornar mais fácil o seu manuseamento com base em índices do *array*.

Os dois programas com as diferenças acima descritas, foram concebidos de forma a comparar as vantagens e desvantagens de cada um.

Memória ocupada	Megas (mb)	
	HammerShark_PICS	HammerShark_ImagList
Antes programa iniciar	19.86	21.93
Menu main	28.46	33.80
Após visualização Nogaro	28.73	43.86

Tabela 6 - Comparação entre os 2 programas.

Os programas apresentam algumas vantagens, um em relação ao outro. Na tabela 5, a linha “após visualização Nogaro” refere-se a um ficheiro de dados relativo à prova do ICARO, de 2006, em Nogaro - França.

Em qualquer um deles, uma imagem só é alterada quando necessário, de forma a haver menos processamento, pois existem 21 *pictureboxs* que podem ser modificadas consoante os dados.

O programa HammerShark_PICS ocupa menos memória que o HammerShark_ImageList¹, tendo a vantagem de praticamente não aumentar de tamanho, com a utilização, ao contrário do programa baseado em ImageList.²

Além disso, comparando as mesmas imagens trabalhadas pelo *Image_List* e colocadas em *picturebox* sobrepostas, têm melhor definição, pois apresentam-se menos *pixilizadas*.

A única vantagem do programa baseado em Imagelist é os tempos de execução no arranque do programa. Os referidos tempos são inferiores aos do HammerShark_PICS, pois este último precisa de carregar todas as imagens. Quando o menu main aparece no ecrã já todas as imagens foram carregadas.

¹ O *Palmtop/PDA* possui uma memória limitada a 60 mb para a execução dos programas, por isso é vital que o programa ocupe pouco dessa memória.

² Existe um “bug” no Compact Framework, referido com mais detalhe no anexo A. Este consiste na não libertação de recursos quando se altera uma imagem de uma *picturebox*. Estes recursos permanecem ocupados até ao limite imposto pelo SO de utilização de memória, de forma a não atingir o nível de *OutOfMemory*, embora este possa ser despoletado se se estiver a acompanhar os erros emitidos pelo programa.

Conclusões

O programa, neste momento, realiza todas as funções pretendidas. Torna visível a informação ao piloto, grava-a num ficheiro em suporte SD-CARD, processa os dados provenientes da ligação bluetooth e estabelece esta ligação automaticamente.

É ainda de referir que foram adicionadas mais algumas potencialidades ao programa, com o intuito de o tornar mais eficiente. Entre as “novas” funções destacam-se o aviso para quando a bateria está num nível baixo, um menu para *replay* de provas, um ficheiro de logging de erros e de estado ao longo da utilização do programa, um menu de configuração automática de bluetooth (criando um ficheiro de configuração_bluetooth para posterior ligação, além de reconexão bluetooth em caso de perda.

A topologia da rede foi alterada, deixando o *palmtop/PDA* de desempenhar ter um papel crucial na rede, passando a ser um visualizador e *logging* de dados. Foram feitas experiências com todos os tipos de rede possíveis, escolhendo-se a melhor e que mais se adequava a cumprir o objectivo pretendido.

No que diz respeito ao *interface* gráfico, também foram testadas as diferentes formas de criação do programa, tendo-se optado pelas imagens sobrepostas.

O programa, na sua totalidade, revela-se robusto, não perde tramas da ligação bluetooth, é rápido durante a sua execução, uma vez que todas as imagens são adicionadas no princípio do programa, garante-se assim maior rapidez de execução.

Em síntese, o programa de *palmtop/PDA* encontra-se finalizado, faltando apenas mencionar e discutir a electrónica de suporte de envio de tramas para o *palmtop/PDA*. Tal informação será analisada no próximo capítulo.

Capítulo 6

Módulo GatewayPalm

Aspectos Gerais

Devido à impossibilidade dos módulos bluetooth conseguirem estabelecer ligações que não sejam ponto-a-ponto (criar *piconets*) e como o *palmtop/PDA* apresentava uma capacidade limitada para o tratamento das tramas e respectivo direccionamento do volante para o módulo de controlo, fez-se uma nova placa que terá o papel de *sniffer* da rede CAN e criar as tramas para respectivo envio para o bluetooth.

Além disso, para efectuar testes, foi adicionada também uma porta *usb-série*.

Introdução teórica

Rede CAN

CAN ou CAN-BUS apresenta-se como a abreviatura de Controller Area Network e define-se como um protocolo para comunicação entre múltiplos dispositivos. Foi criada pela Bosch, em 1991, para uso na indústria automóvel, nomeadamente para a ligação dos controlos nos motores de automóveis. Actualmente é utilizada em diversas áreas e define apenas a ligação física e o controlo do acesso ao meio de transmissão (ligação de dados).

Alguns aspectos da rede CAN:

- *Multi-master, broadcast, serial bus*;
- Taxa de transmissão de 5kBit/s a 1 Mbit/s;
- O tamanho do barramento depende da taxa de transmissão, que é de aproximadamente de 40 m @ 1 Mbit/s, 1000m @ 5 Kbit/s;
- Codificação dos bits: NRZ - *Non return to zero*;
- Tx/Rx sincronização, usando *bit stuffing* (5 bits);
- Acesso ao barramento assíncrono;
- *Data* entre 0 -8 bytes;
- Identificadores com 11 bits, na versão A, e 29 bits na versão B.

As versões acima referidas estão esquematizadas na Figura 56, onde é possível ver os campos da trama completa. É possível utilizar as duas versões (designadas por vezes como *standard* ou *extended*) em simultâneo, desde que:

- Se utilizem ou controladores CAN2.0B ou controladores designados por CAN2.0B passivos;
- Um controlador CAN2.0B passivo detecta as tramas 2.0B pela verificação do *bit* de IDE (*Identifier Extension*), ignorando-as;

- Um controlador 2.0A que, sem esta funcionalidade, reagirá a uma trama 2.0B com tramas de erro.

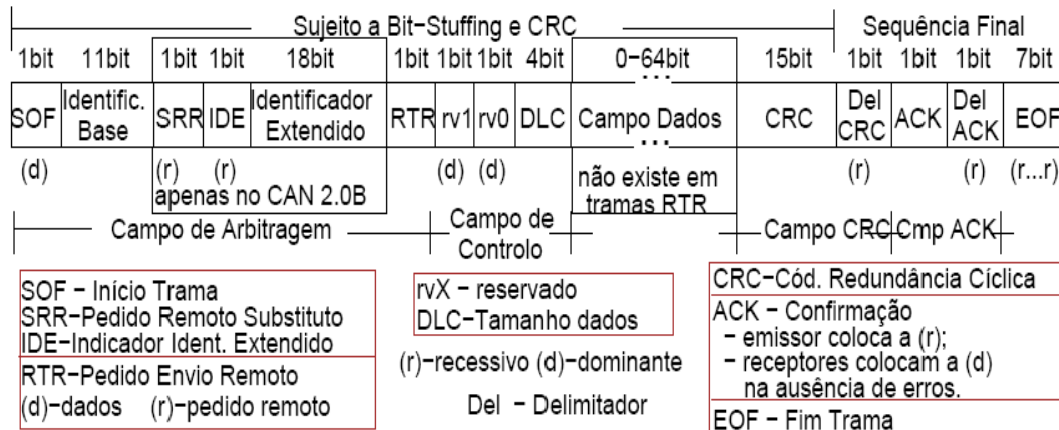


figura 56 - Trama CAN.

A camada física do CAN força a sincronização de todos os nós da rede e faz com que haja apenas um único *bit* em qualquer altura no barramento.

A codificação é feita através de *bits* dominantes (0) e recessivos (1). Um *bit* recessivo só aparece no barramento quando todos os nós transmitem *bits* recessivos, bastando que apenas um nó transmita um *bit* dominante para que este prevaleça. Cada nó espera que a CAN esteja inactiva para começar a transmitir uma nova trama, podendo haver vários nós a iniciar a transmissão simultaneamente.

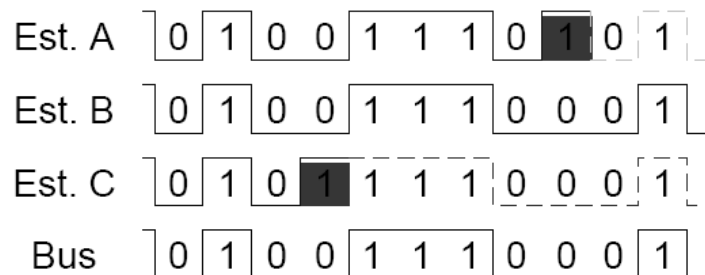


figura 57 - Prioridade sobre o barramento.

Cada mensagem possuiu o seu próprio identificador - quanto menor este for, maior será a prioridade da mensagem. Em caso de transmissão simultânea, cada nó tentará transmitir a sua mensagem até que, durante a transmissão do identificador, a mensagem com maior prioridade (com o menor identificador) se imponha sobre as outras. Após a transmissão do identificador, apenas um nó estará a transmitir, com excepção para o caso da mensagem ser um pedido de transmissão remota (único caso em que vários nós podem transmitir simultaneamente).

GatewayPalm - Hardware

Em termos de *hardware*, a placa apresenta como componentes principais:

- PIC18F458;
- CAN TRANSCEIVER;
- O módulo bluetooth [19], através de um conector, sendo colocada a placa via bluetooth no conector.

A esse conector também se pode ligar um Mini USB-Serial UART Module, para o envio de tramas, bem como testes, através da porta serial. É de salientar ainda que existe mais electrónica de suporte, nomeadamente os reguladores de 12V para 5V, além do regulador de 5V para 3.3V, o RJ45 para programação, entre outros.

No que diz respeito ao esquema eléctrico e de *layout* da placa, o esquema encontra-se no anexo C.

GatewayPalm - Software

Existe um evento, o RX_CAN, que indica que foi recebida uma trama CAN. A partir desse momento, é despoletado um evento RX_CAN_DATA, que processa a informação vinda da CAN, processando qual a placa que enviou a informação e o sinal a que se refere. A partir desse momento, a informação é adicionada a um *buffer* até se atingir o comprimento máximo de 70 bytes para uma trama, ou seja, uma trama contém vários sinais de maneira a reduzir o *overhead*. Quando esse comprimento é atingido, adicionam-se 2 bytes de CRC e enviam-se para *logging*, através do bluetooth, para o *palmtop/PDA* a informação.

Além disso do evento já referido, existe um outro evento denominado EV_TICK_QUART_SEC: que, de 250 em 250 ms, envia informação para o *palmtop/PDA* – caso exista, mesmo que não tenha os 70 bytes.

Aspecto das tramas

00000000	Header 0
1NNNNNNN	Header 1 (Num. De bytes do restante da trama < 72)
BBBSSSSS	BBB = Board SSSSS = Signal ID
DDDDDDDD	Data
.....	
BBBSSSSS	
DDDDDDDD	Em caso de dados multi-byte
DDDDDDDD	Adoptar formato big-endian
CCCCCCCC	CRC 0
CCCCCCCC	CRC1

Testes efectuados

Foi possível observar que o envio do comando `screen_change` do volante para o *Palmtop/PDA*, através do Módulo de Controlo e passando pela rede CAN até ao GatewayPalm, era feito de forma “instantânea”, sendo visível no ecrã do *palmtop/PDA* a troca efectuada, não havendo assim a possibilidade de atrasos.

Outro teste realizado consistiu, essencialmente, em ligar o PC ao Módulo de Controlo, via bluetooth, e enviar 1000 tramas de 8 *bytes*. No *palmtop/PDA* verificou-se que não se perdiam tramas.

Conclusões

O módulo responsável pela captação das tramas na rede CAN consegue enviá-las para o *palmtop/PDA*.

Em relação à placa, esta é composta pela placa *sniffer* de rede CAN, tendo um conector onde é possível colocar uma *serial port* (mini USB) ou um módulo bluetooth, encontrando-se ambos funcionais.

Deste modo, foi possível mudar a topologia de rede existindo um programa de *Palmtop/PDA* funcional e uma nova placa, também funcional, que envia os dados para este.

Capítulo 7

Conclusões gerais

O desenvolvimento deste trabalho consistiu na criação e alteração de diversos módulos para a teleoperação de um veículo. Analisando cada módulo em detalhe conclui-se que, em relação ao módulo de visão remota, embora inicialmente se pensasse que seria algo mais simples, uma vez que se pretendia um sistema mais robusto e com menos lixo no espectro, o sistema tornou-se mais complexo e bastante mais trabalhoso. Foi, também, necessário reformular os cálculos teóricos para algo que, de facto, se enquadrava na realidade.

No que diz respeito ao módulo volante, este sistema já se encontrava funcional, tendo sido apenas necessário fazer alguns ajustes para a sua total integração no sistema global ICARO. Relativamente à utilização do giroscópio, não foi possível colocá-lo a funcionar correctamente, tendo-se optado pela sua não integração no volante.

A electrónica de módulo de controlo/GatewayPalm foi, por sua vez, alterada, deixando de ter as duas funções especificadas no ano transacto passando, agora, a receber os comandos do volante, a retransmiti-los para os módulos CAN e a fazer o controlo - quando em condução remota. Esta placa, embora funcional, poderia ser melhorada, através da alimentação dos servos directamente da placa. Esta modificação permitiria a não existência de fios soltos mas sim de conectores, através de reles que (des)activariam os motores.

Relativamente à criação do software do *palmtop/PDA* e à electrónica a ele associada, o sistema mostrou-se funcional, necessitando apenas de pequenos ajustes relacionados com o facto da corrida se desenrolar em Hockenheim, Alemanha, no ano de 2009. Tal situação obriga a ajustar os parâmetros usados para a distância total de pista, o número de voltas e o tempo total de prova que, neste momento, estão definidos para a prova de Nogaro, França. Estas alterações centrar-se-ão, essencialmente, na óptica do utilizador, uma vez que o *palmtop/PDA* recebe e processa dados do veículo, correctamente. A nova topologia de rede provou ser eficaz, pois possibilita a não existência de uma *gateway* entre o volante e o veículo existindo, agora, 2 ligações ponto-a-ponto. A nova placa GatewayPalm têm a função de sniffer da rede CAN e retransmite, por bluetooth, para o *palmtop/PDA*, provando ser uma boa metodologia a utilizar.

Numa perspectiva de trabalho futuro, embora a placa do módulo de controlo possa também ser incluída, esta não é prioritária pois, além do giroscópio, existe um tópico da proposta de trabalho inicial que não foi possível desenvolver neste projecto. Parafraseando, o “desenvolvimento de algoritmos baseados em processamento digital de imagem com vista à integração de mecanismos automáticos de suporte à segurança, nomeadamente em situações de falha de comunicação entre a box e o veículo”, servirá como uma primeira etapa para a condução autónoma do veículo.

Referências

- [1] – Daniel Castanheira, “Teleoperação de um veículo de eco maratona”, IEETA, Universidade de Aveiro 2007.
- [2] - http://www.jneuhaus.com/fccindex/470_mhz.html
- [3] - <http://wireless.per.nl/reference/chaptr03/pel/pel.htm>
- [4] - ANDERSON ESCUDERO , “Modelagem para o cálculo da perda no percurso de propagação em áreas residenciais nas faixas de VHF e UHF “, Instituto Nacional de Telecomunicações, 2004.
- [5] – Televes: Datasheet 5522, Televes 2007.
- [6] - <http://maps.google.com/>
- [7] - Neves, Apontamentos da disciplina de Propagação e Radiação de Ondas Electromagnéticas, DETI, Universidade de Aveiro 2007.
- [8] – Philips Semiconductors: Datasheet BLF542 , Philips 2003.
- [9] – Murata: Datasheet Piezoelectric Vibrating Gyroscope,Murata 2006.
- [10] - <http://tom.pycke.be/mav/70/gyroscope-to-roll-pitch-and-yaw>
- [11]http://www.killerbees33.com/index.php?option=com_content&task=view&id=84&Itemid=50
- [12] - Microchip: Datasheet PIC18F458, Microchip 2006.
- [13] – KONDO – Datasheet KRS-4014, Kondo 2007.
- [14] - <http://franson.com/bluetoools/>
- [15] - <http://labs.developerfusion.co.uk/convert/csharp-to-vb.aspx>
- [16] - <http://www.homeandlearn.co.uk/net/nets9p7.html>
- [17]<http://blog.opennetcf.org/ctacke/PermaLink,guid,987041fc-2e13-4bab-930a-f79021225b74.aspx>
- [18] - <http://forums.microsoft.com/MSDN/ShowPost.aspx?PostID=2566824&SiteID=1>
- [19] – Free2move: Bluetooth Module, Datasheet F2M03GLA , Free2move 2007.
- [20] – Microsoft : Visual Basic .NET Passo a Passo , Microsoft 2007.

Anexo A

1. *Creating a bitmap using the stream constructor will construct a DIB (Device Independent Bitmap).*
2. *Creating a bitmap using the width/height constructor will construct a DDB (Device Dependent Bitmap).*
3. *DIB's are allocated out of the virtual address space of the application.*
4. *DDB's are allocated by the driver. This typically means that they are allocated in the virtual address space of gwes.exe. Alternatively, the driver could allocate these in dedicated video ram.*
5. *Creating a bitmap with the stream constructor will generate a fair amount of garbage as it copies data from one buffer to the other.*

When we perform a GC because of an OOM in the stream constructor case, we will almost certainly have some amount of garbage that we can free back to the OS immediately. This will also trigger the finalizer to run on another thread as soon as possible. That should help the next call to bitmap creation.

When we perform a GC because of an OOM in the width/height constructor case, it is fairly likely that the OOM is caused because of virtual memory exhaustion in gwes.exe. Thus freeing memory in our process will not help the memory condition in gwes.exe. We need the bitmap finalizer to run before this would actually free memory in a way that would help this scenario. While the finalizer thread would certainly have been triggered to start, it most likely will not get a chance to free bitmaps before we OOM while trying to allocate a bitmap immediately after triggering a GC on the initial thread.

In short, we have 2 different types of Bitmap in our runtime with varying performance and allocation characteristics. DDBs are generally faster to manipulate and draw to the screen than DIBs, but they are constructed in an external memory space that can cause allocation confusion and cause the performance of calls to LockBits or Save to be slow. If a DIB is desired and you wish to construct it based on width and height, we provide a function that constructs a Bitmap with a width, height, and pixelformat specified. This function will construct a DIB instead of a DDB.

Em suma, quando se utilizam pictureboxs para ir mudando constantemente imagens, os recursos não são libertados.

Existem 2 possíveis técnicas para contornar este problema:

No final da utilização da picturebox fazer um

`GC.WaitForPendingFinalizers()`

`GC.Collect()`

Que força o programa a esperar por eventos que ainda não tenham sido finalizados, seguido de um Garbage Collect que fecha todos os recursos que não estejam a ser utilizados.

Porém apresenta alguns problemas nomeadamente que no caso em que tenhamos um programa (como é o caso do capítulo 5) que têm múltiplos eventos que podem ser chamados podemos correr o risco de eventualmente eliminar recursos utilizados por eventos que realmente precisam deles, o que vai originar instabilidade no programa e processamento de informação.

A outra técnica é:

`Picturebox.image.dispose()`

`Picturebox.image = nova_imagem`

Ou seja apagar a imagem antiga e só depois colocar a nova, não deixando a cargo do compilador essa tarefa e sim forçando-o a isso. Porém esta vertente não é 100% fiável, pois os recursos continuam a aumentar (a um nível muito inferior ao que se regista caso não se faça o dispose da imagem) o que se torna mais evidente em programas como o desta tese em que existem cerca de 20 pictureboxs que podem ser alteradas.

Anexo B

Mensagens RF

Existem três ligações sem fios, uma que comunica com o PDA, outra que permite a comunicação entre volante -modulo de controlo e outra ligação que permite a comunicação com as boxes.

Tramas	
00000000	Header 0
1NNNNNNN	Header 1 (Num. De bytes do restante da trama < 72)
BBBSSSSS	BBB = Board SSSSS = Signal ID
DDDDDDDD	Data
.....	
BBBSSSSS	
DDDDDDDD	Em caso de dados multi-byte
DDDDDDDD	Adoptar formato big-endian
CCCCCCCC	CRC 0
CCCCCCCC	CRC1

IDs das mensagens circulantes nas ligações sem fios

Placa	Board ID
Gateway Palm	0
GPS	1
Roda	2
Motor (monitorização)	3
Motor (controlo)	5
Volante	6

Placa Gateway Volante (Board ID 0)

ID do sinal	Descrição	Nº de bytes	Frequência (n/Seg.)	Nº de bytes/Seg.
2	Start/Stop	1	Esporádico(4)	N.A.
3	Buzina	1	Esporádico(4)	N.A.
4	Acelerador	1	Esporádico(20)	N.A.
5	Screen Change	1	Esporádico(4)	N.A.
6	Temp. habitáculo	1	1	1
7	SetPoint acelerador	1	Esporádico	N.A.
8	Reset Roda	0	Esporádico	N.A.
9	Brake	1	20	20
10	Steer	1	20	20
11	Remote	0	Esporádico	N.A.
Totais		8		40 a 74

Start/Stop	Aa	Aa = 0x00 -> Stop pressed aa = 0xFF-> Start pressed
Buzina	Aa	Aa = 0x00 -> OFF aa= 0xFF -> On
Acelerador	Aa	Aa= posição do acelerador em inteiro (0 – 99=
Screen Change	Aa	Aa= 0xFF
Temp. habitáculo	Aa	*por de_nir (devolve 0x00 de momento)
Setpoint acelerador	Aa	aa - setpoint do acelerador em inteiro (0 - 99)
Brake	Aa	aa - setpoint do travão em inteiro (0 - 99)
Steer	Aa	aa - setpoint da direcção em inteiro, graus, (-45 - 45)

Placa GPS (Board ID 2)

ID do sinal	Descrição	Nº de bytes	Frequência (n/Seg.)	Nº de bytes/Seg.
1	Latitude	4	1	4
2	Longitude	4	1	4
3	Altitude	2	1	2
4	Velocidade	2	1	2
5	Direcção	2	1	2
6	Diluição de precisão	2	1	2
7	Hora	3	1	3
8	Data	3	1	3
9	Dados válidos	1	1	1
Totais		23		23

Latitude	Dd0,mm,mm1,mm2
Longitude	Dd1,mm,mm1,mm2
Altitude	Aaaa
Velocidade	Vvvv
Direcção	Dddd
Diluição de precisão	Pppp
Hora	Hh , mm , ss
Data	Dd , ms, yy
Dados validos	V

Legenda da tabela anterior:

- dd0 - número de graus (0 - 90)
- dd1 - número de graus (0 - 90)
- hh - hora em inteiro (0 - 23)
- mm - número de minutos (0 - 59)
- ss - segundos em inteiro (0 - 59)
- mm1 - número de centésimas de minutos (0 - 99)

- mm2 - número de decimas de milésimas de minuto (0 - 99)
- aaaa - altitude em decímetros (-32768 - 32767)
- vvvv - velocidade em décimos de nó (0 - 18518)
- dddd - direcção em décimos de grau (0 - 3599)
- pppp - diluição de precisão em decimas (0 - 999)
- dd - dia em inteiro
- ms - mês em inteiro
- yy - ano em inteiro
 - v - 0 - invalido; 1 – valido

Placa Motor Monitorização (Board ID 3)

ID do sinal	Descrição	Nº bytes	Freq.(n/Seg.)	Nº bytes/Seg.
1	Velocidade	2	4	8
2	Distância	2	2	4
3	Tempo de prova	2	1	2
4	Válvula de combustível	1	1	1
5	Tensão da sonda Lambda	1	2	2
6	Temperatura do ar de admissão	1	1	1
7	Temperatura do Óleo	1	1	1
8	Pressão do combustível	1	1	1
9	Tensão da bateria de controlo	1	1	1
10	Pressão na Câmara de admissão	1	1	1
11	Tensão na Bateria de potência	1	1	1
12	RPMs	2	1	2
13	Total de combustível	3	1	3
14	Posição do acelerador	1	1	1
Totais		20		29

Velocidade	IMP(6bits) + WIN(10bits)	IMP - Número de impulsos WIN - Janela temporal em múltiplos de 250µs
Distância	N	Valor dado por $D = \frac{345 \cdot N}{1000}$
Tempo de prova	T	T - Número de segundos de prova
Válvula de combustível	F	F = 0 – Válvula de combustível fechada F = 1- Válvula de combustível aberta
Tensão da sonda Lambda	Lambda	$V_{\lambda} = \frac{5}{4} \cdot \frac{\text{Lambda}}{255}$

Temperatura do ar de admissão	T_{Ar}	$T = \frac{67.582 \cdot T_{Ar}}{255} + 5.194$
Temperatura do Óleo	T_{Oleo}	$- 21 \cdot \ln \frac{18 \times 10^3 \cdot \frac{T_{Oleo}}{255} - 380}{25500}$
Pressão do combustível	P_{Comb}	$P_{COMB} = \left(5 \cdot \frac{P_{Comb}}{255} - 0.5 \right) \cdot bar [abs]$
Tensão de bateria electrónica	$VBATT$	$V_{batt} = 14.803 \cdot \frac{VBATT}{255} (V)$

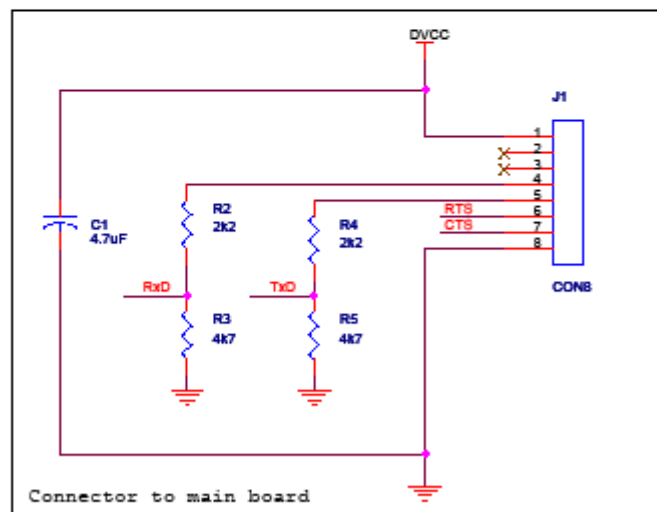
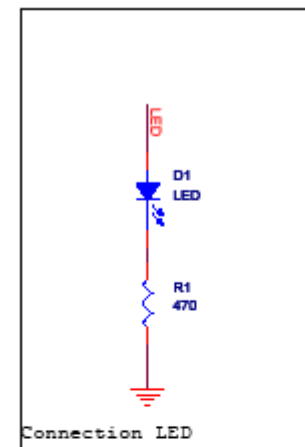
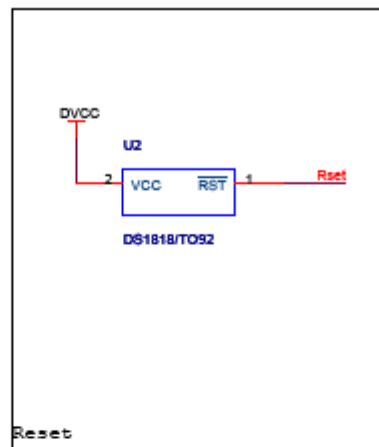
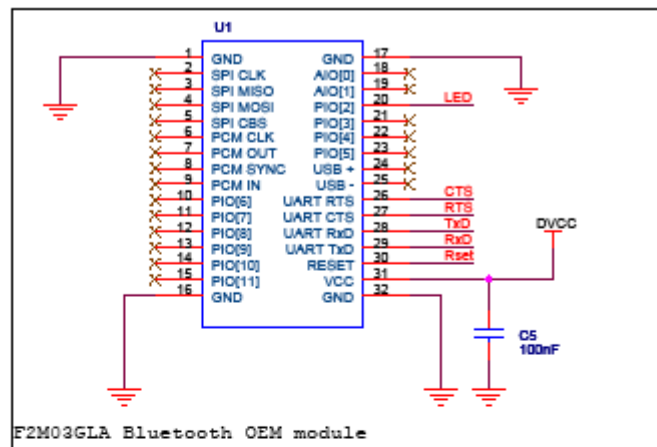
Anexo C

Layouts e esquemas eléctricos

Todas as medidas de dimensões das placas estão em milímetros.
É apresentado os esquemas eléctricos e os layouts das placas (versões finais).

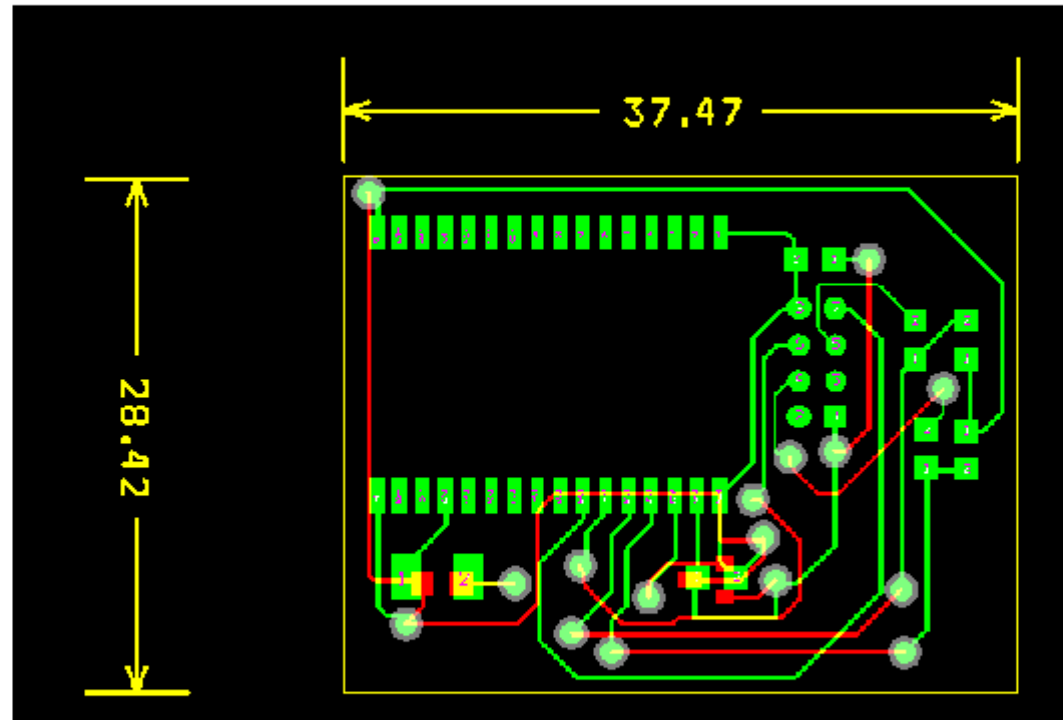
Ordem de aparecimento das placas neste anexo:

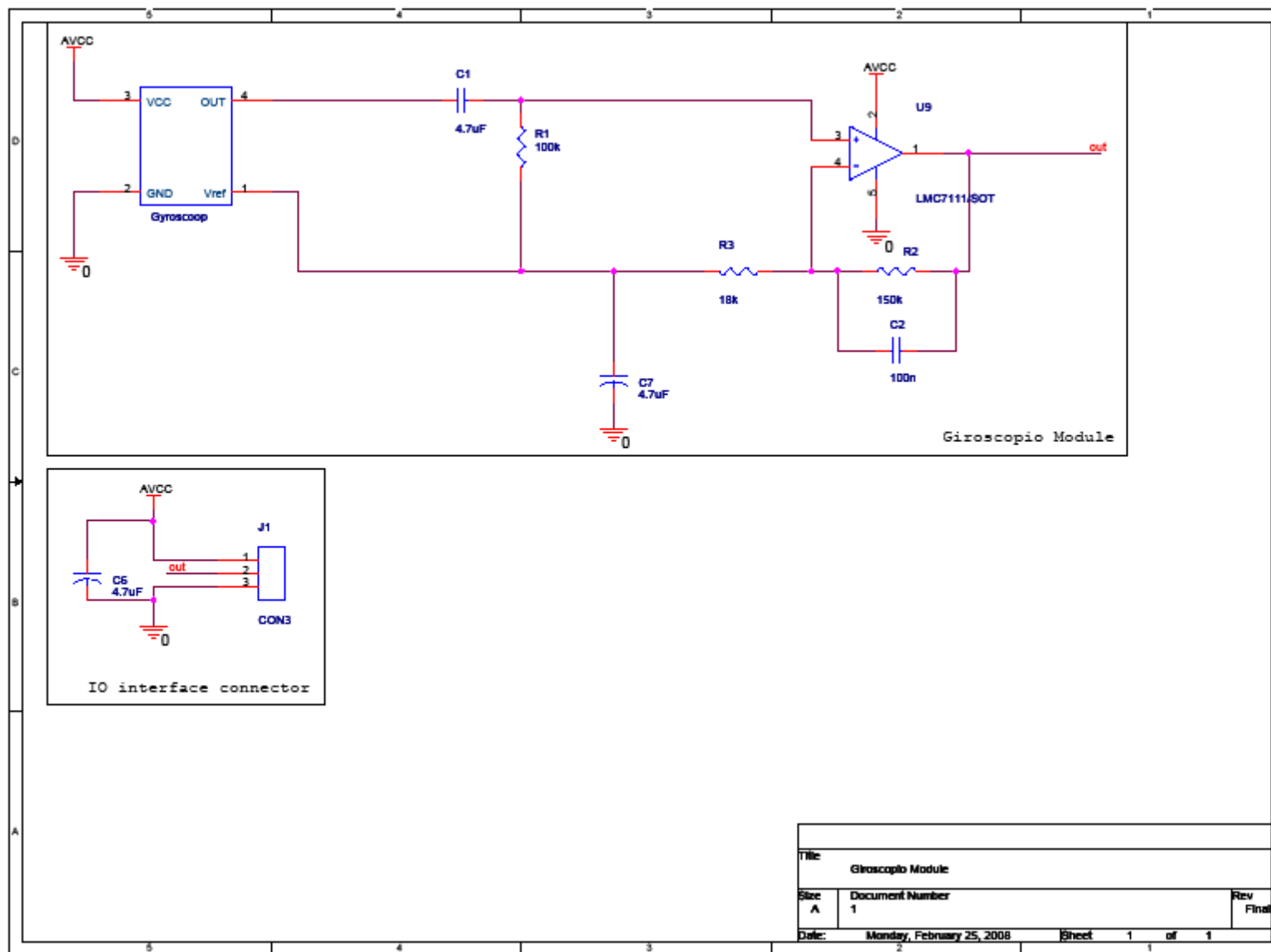
- 1º Conector Bluetooth para a placa GatewayPalm
- 2º Giroscopio
- 3º Placa GatewayPalm
- 4º Filtro UHF
- 5º Conector USB para a placa GatewayPalm



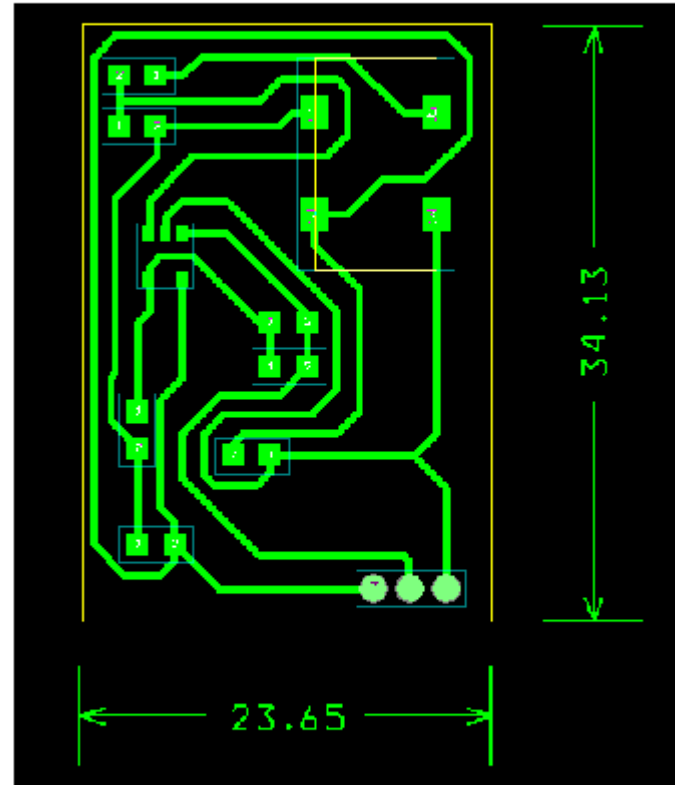
Rev1: Colocação de divisor resistivo- 5V -3.3V

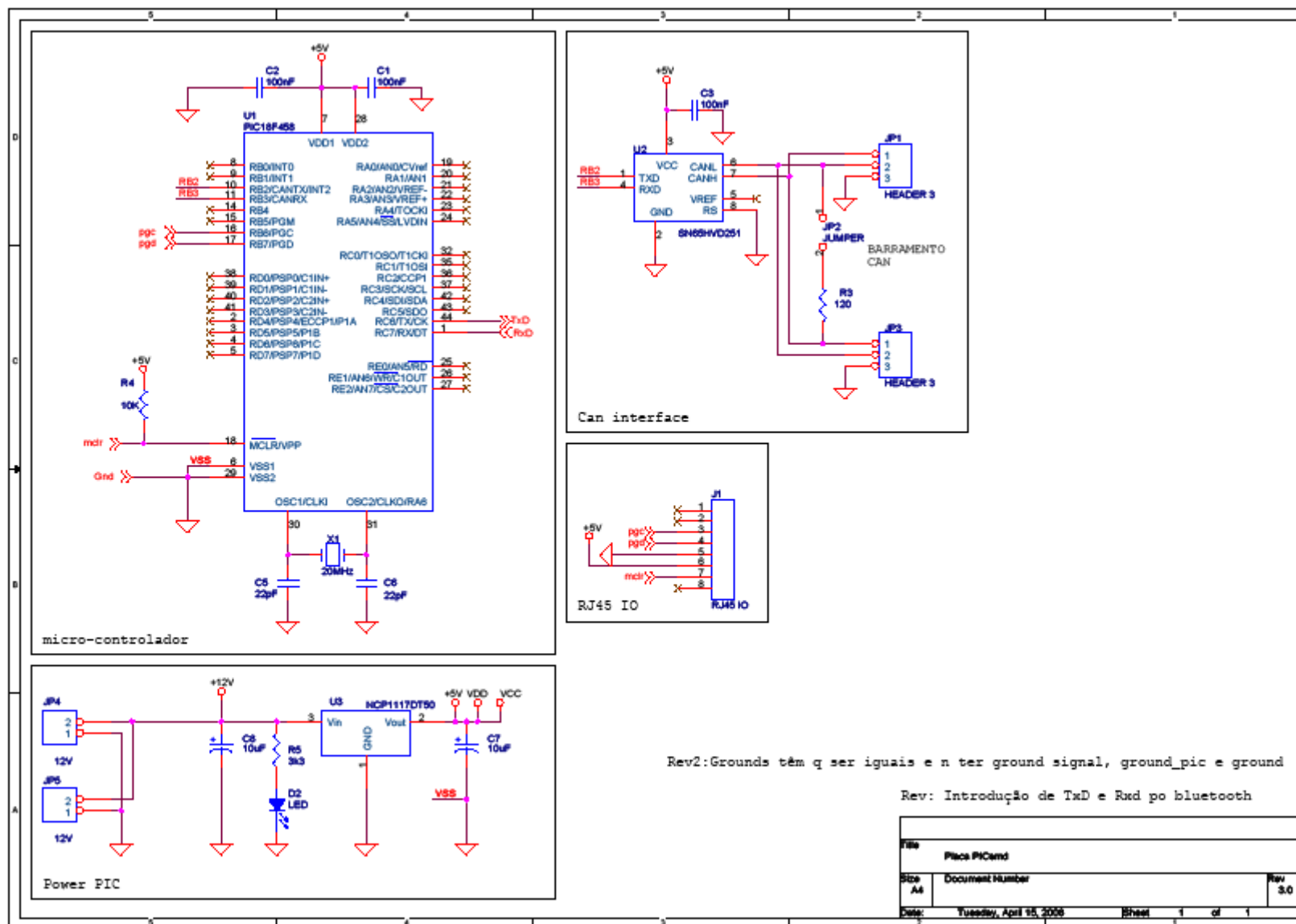
Bluetooth Piggy Free2Move		
Title:	Bluetooth for PutTmeter	
Size A	Document Number	Rev
	BTP 1.1	Final
Date:	Monday, April 14, 2008	Sheet 1 of 2

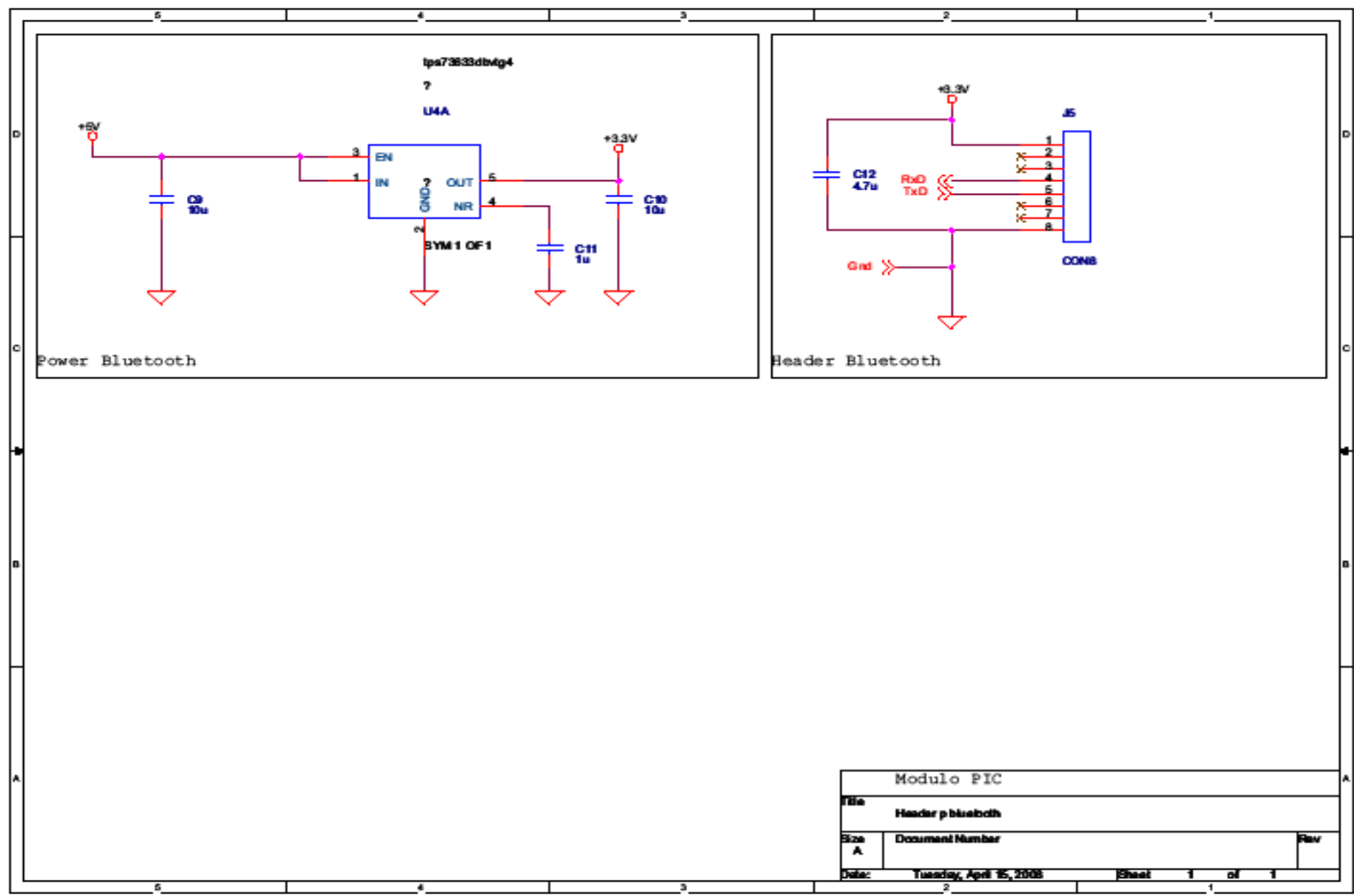




Title: Gyroscopic Module		
Size A	Document Number 1	Rev Final
Date: Monday, February 25, 2008 Sheet 1 of 1		



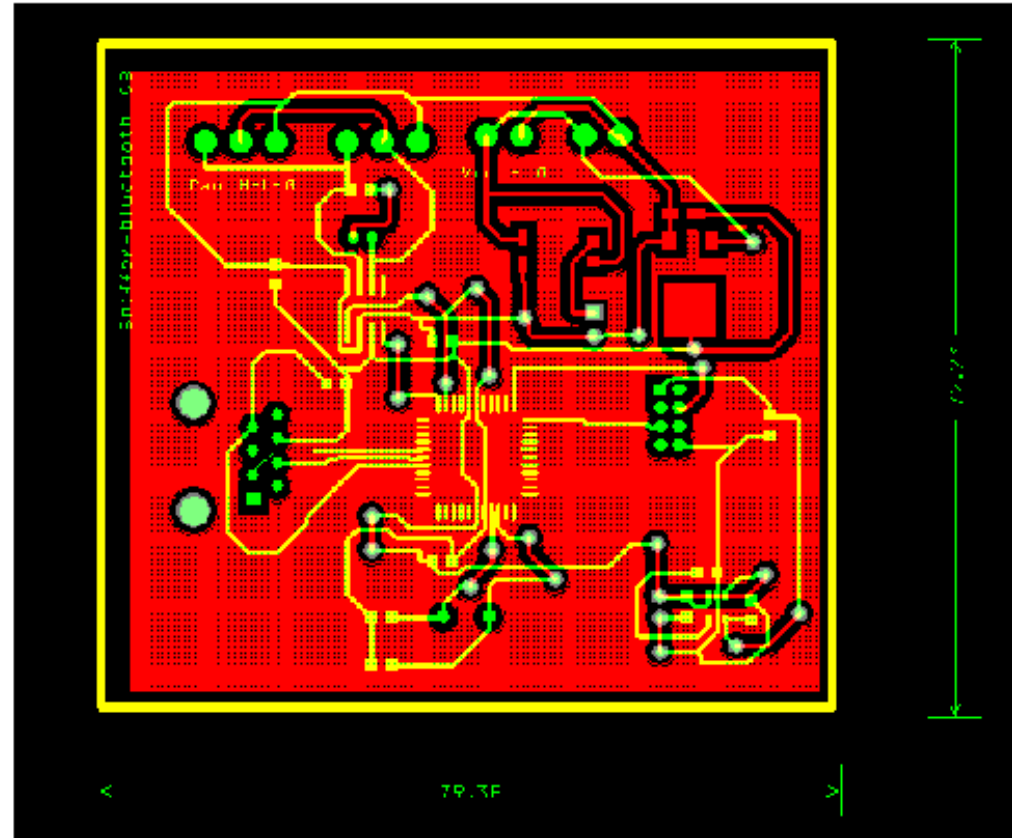


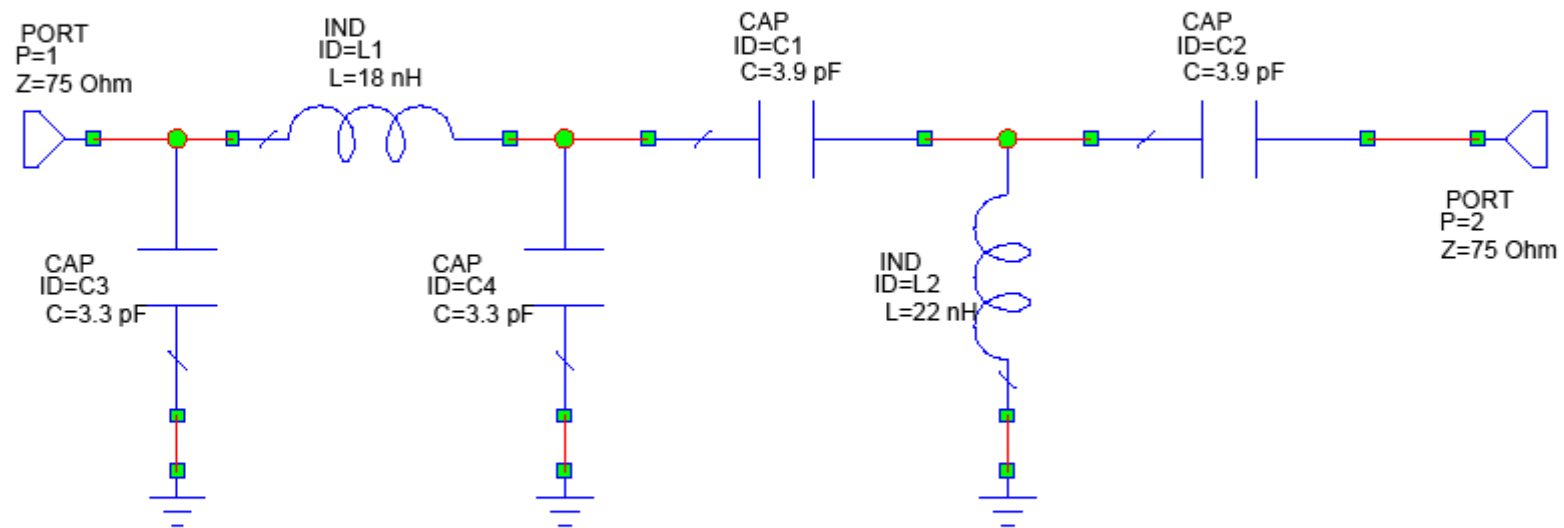


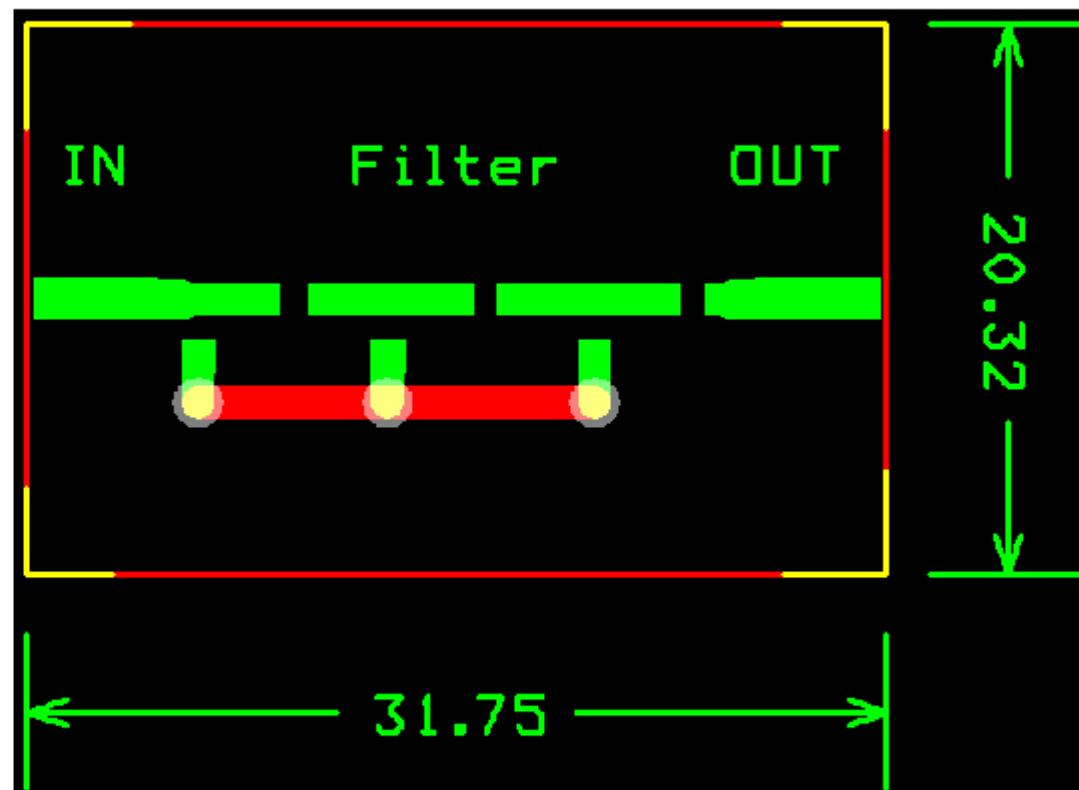
Power Bluetooth

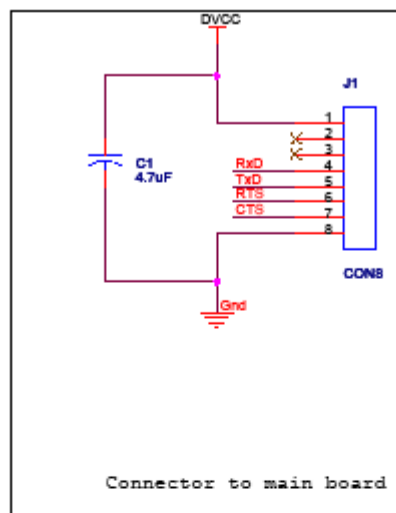
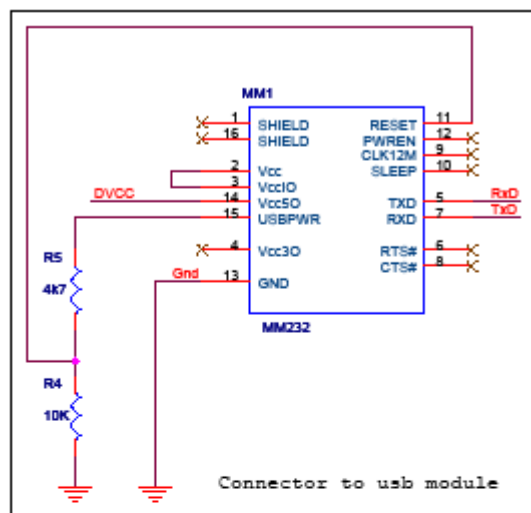
Header Bluetooth

Modulo PIC		
Header p bluetooth		
Size A	Document Number	Rev
Date:	Tuesday, April 15, 2008	Sheet 1 of 1









USB- Serial 232		
Title	USB-Serial to Sniffer	
Size	Document Number	Rev
A	BTP 1.1	Final
Date:	Friday, April 11, 2008	Sheet 1 of 2

